LiNUX+

+DVD MOLINUX 4.2 | FREEBSD 7.1 LA PRIMERA DISTRIBUCIÓN GRANDE DEL AÑO 2009

LiNUX+ LINUX+DVD Nº 52

LiNUX+

LA MAYOR REVISTA EUROPEA SOBRE LINUX

PRECIO 8,50 EUR Nº 52 MENSUAL ISSN 1732-7121

DVD

HAZLO MÁS FÁCIL DE ADMINISTRAR Y MÁS LLAMATIVO PARA LOS USUARIOS

CREA TU PROPIO FORO EN PHP

GAMBAS

LA PRIMERA PARTE DEL NUEVO CICLO DE ARTÍCULOS SOBRE ESTE LENGUAJE DE PROGRAMACIÓN

FINANCIACIÓN DE PROYECTOS DE SOFTWARE LIBRE

¿QUIÉN PAGA POR TODO ESTO?

EASY PEASY

EXPRIME TODO EL POTENCIAL DE TU EEE PC

SCRIBUS

SOFTWARE DE AUTOEDICIÓN

LENGUAJES, LENGUAS Y DIALECTOS

CONVIÉRTETE EN UN POLIGLOTA DE PROGRAMACIÓN

NHOPKG

PROYECTO DE JOTAHACKER QUE REVOLUCIONARÁ EL MUNDO DE SOFTWARE LIBRE

MOLINUX 4.2

LA NUEVA ENTREGA DE UN SISTEMA OPERATIVO CREADO POR LA JUNTA DE COMUNIDADES DE CASTILLA



FREEBSD 7.1

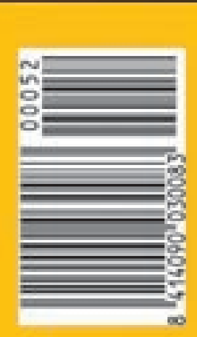
LA PRIMERA DISTRIBUCIÓN GRANDE DEL AÑO 2009



SÓLO AQUÍ

LIBERA TUS OÍDOS

GUÍA POR LA MÚSICA LIBRE EN LA RED



virtualpyme

El Enclave Óptimo para su Web

SERVICIOS DE ALOJAMIENTO WEB COMPARTIDO

PLANES	Hospedaje Mínimo	Hospedaje Básico	Hospedaje H100	Hospedaje Ecom
Espacio en disco (Mb)	1.000 Mb	2.000 Mb	3.500 Mb	5.000 Mb
Transferencia de datos (Mb/mes)	5.000	10.000	15.000	20.000
Cuentas POP/IMAP de correo	10	40	250	1.000
Dominios GRATUITOS incluidos	1	1	2	2
Precio	**3,65 €/mes.**	**6,04 €/mes.**	**11,91 €/mes.**	**16,61 €/mes.**

SERVIDORES DEDICADOS DE ÚLTIMA GENERACIÓN

<table>
<tr><th>GAMA BÁSICA</th><th>GAMA MEDIA I</th><th>GAMA MEDIA II</th><th>GAMA SUPERIOR</th></tr>
<tr><td>Intel Quad Core 2.4 Ghz</td><td>Dual-Core Xeon 3.2 Ghz</td><td>Intel Quad Core 2.4 Ghz</td><td>Intel Quad-Core Xeon 2.5 Ghz</td></tr>
<tr><td>1 Gb ECC Ram</td><td colspan="2">2 Gb ECC Ram</td><td>4 Gb ECC Ram</td></tr>
<tr><td>SATA-II 250 Gb</td><td colspan="2">2 HD SATA-II 500 Gb Hot plug</td><td>2 HD SAS 147 Gb Hot plug</td></tr>
<tr><td>1.000 Gb de Tx / mes</td><td colspan="2">1.500 Gb de Tx / mes</td><td>2.000 Gb de Tx / mes</td></tr>
<tr><td>95,00 €/mes.</td><td>128,24 €/mes.</td><td>138,24 €/mes.</td><td>233,21 €/mes.</td></tr>
</table>

* La transferencia de datos incluye una garantía de disponibilidad del 100% del tiempo bajo contrato SLA

virtualpyme

Es una empresa especializada en ofrecer servicios y soluciones de alojamiento web y registro de dominios. En sus seis años de existencia hemos obtenido la confianza de más de 6.500 clientes con alrededor de 10.000 servicios activos y más de 70.000 dominios registrados entre las distintas extensiones.

Tels.: 902 52 02 52 · 913 155 586 · info@virtualpyme.es · www.virtualpyme.es



Borges y LiNUX+

El mejor profesor es éste que ha aprendido mucho y que sigue aprendiendo. Lo hace para sí mismo, pero aprovechan sus estudiantes. Siempre buscaba esta clase de profesores: enfocados en su propio desarrollo, que eran capaces de mostrar interés hacia lo que explicaban y que sabían *demasiado*. Por otra parte, me gustaban aquellos a quienes no les asustaba decir *no lo sé, pero si deseáis, lo voy a comprobar*. Les tenía mucho respeto y confianza. Hasta hoy día suelo pensar que ese respeto era recíproco. Porque tener bastante confianza en los alumnos para ser capaz de decirles que no se sabía algo es un gran rasgo de carácter. Ya que todos sabemos qué consecuencias puede tener una revelación de este tipo.

Ya no soy estudiante y no encuentro muchos profesores en mi vida profesional. Pero sí que me encuentro con muchos escritores y periodistas. Autores de libros, artículos, ensayos y folletines. Y como para un buen profesor es indispensable aprender todo el tiempo, para ser un buen escritor sobre todo hay que leer. Dicen que para escribir una buena página de texto hay que leer por lo menos cien páginas escritas por otros.

Nosotros venimos a vuestras casas cada mes con casi cien hojas. No me puedo creer cuánto leen nuestros autores para llenar estas hojas con artículos de calidad.

> “*que otros se jacten de las páginas que han escrito; a mí me enorgullecen las que he leído. Jorge Luis Borges*”

Gracias a Lino García Morales os enteraréis más sobre varios lenguajes de programación (p. 18). Os aconsejo leer este artículo muy detalladamente, ya que estoy segura que os ayudará ordenar todo lo que habéis aprendido hasta ahora sobre programación.

El artículo de Lino os servirá como base teórica para ejercicios prácticos de programación propuestos por David Puente Castro y Jorge Emanuel Capurro. El primero os guiará por la creación del foro en PHP (p. 32) y el último os enseñará con todo detalle la programación con Gambas (p. 22).

Si queréis saber quién es Jotahacker y por qué estamos seguros que dentro de unos años vamos a leer sobre él en cada revista sobre informática y tecnología, leed el artículo de Juan Carlos de la Vega Salamanca (p. 84).

En este número iniciamos la sección *Hacking para Linuxeros*. No es una sección completamente nueva, pero sentimos la necesidad de dividir *Seguridad* en dos secciones distintas: *Seguridad* y la mencionada *Hacking para Linuxeros*. Espero que esta división os resulte más clara y facilitará la lectura de la revista. Este mes en dicha sección os proponemos dos artículos: *Interceptando conversaciones Messenger* por David Puente Castro (p. 50) y *Firewall con 2 canales de internet* por Samuel Diaz (p. 58).

Recuerdo muy bien una frase de Jorge Luis Borges, uno de los escritores más grandes del siglo XX: *que otros se jacten de las páginas que han escrito; a mí me enorgullecen las que he leído*. Que otros se jacten de las páginas que vais a leer este mes en LiNUX+.

Aleksandra Tęsiorowska
Redactora de Linux+

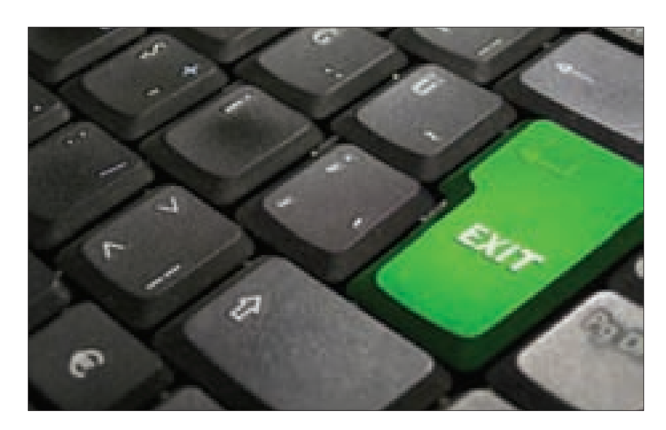


En este número



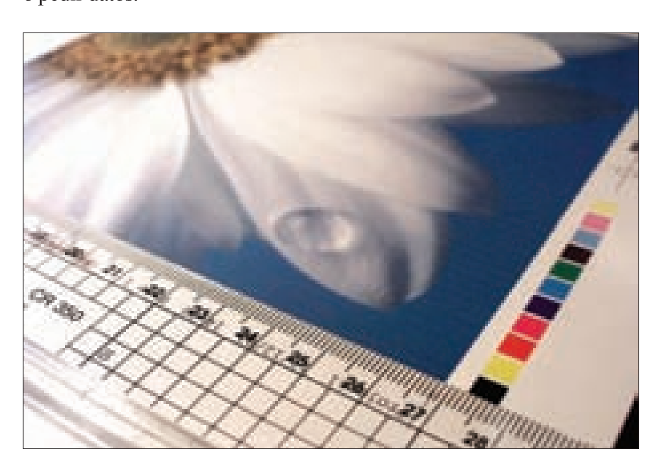
Tema del número Programación

hardware



conocimiento libre

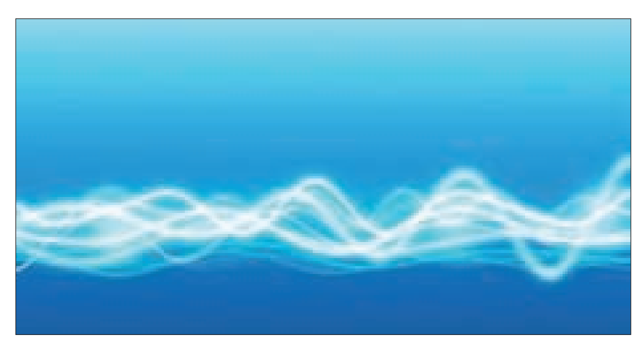
práctica



multimedia



linux en la educación



mundo Linux

Molinux

Molinux es la distribución de GNU/Linux impulsada por la Junta de Comunidades de Castilla-La Mancha a través del Centro de Excelencia de Software Libre de Castilla-La Mancha (CESLCAM), que tiene como fin ofrecer a todos los ciudadanos una plataforma libre con la que poder trabajar, conectarse a Internet, recibir formación y, en general, realizar todas aquellas tareas relacionadas con la tecnología.

Versión 4.2

La distribución Molinux, a día de hoy se encuentra en su versión estable 4.2 y cuenta con paquetes software que cubren todas las necesidades de los usuarios, desde software ofimático como OpenOffice, todas las herramientas para Internet, hasta las herramientas multimedia más conocidas y de mayor calidad, e incluso herramientas de diseño y desarrollo de software.

Molinux también está enfocado a cubrir las necesidades de las empresas, profesionales y administraciones públicas y privadas, ofreciéndoles herramientas administrativas, software de gestión y bases de datos, además de la formación necesaria y un soporte técnico permanente.

La nueva versión de la distribución que está incluida en el presente número de LiNUX+ incluye importantes mejoras en los sistemas existentes, además de diversas novedades en software, documentación, guías, etc. En este sentido, estaban abiertas varias lineas de investigación, a través de las cuales se perseguía ampliar y mejorar los servicios y funcionalidades que ofrecía el sistema como era, por ejemplo, el estudio de las necesidades de diversos sectores (educativos, profesionales, etc.) con el fin de conocer los requerimientos tecnológicos de cada sector y poder ofrecer soluciones verticales.

Características principales

Molinux, a su vez, está basada en la distribución Ubuntu, de la cual se aprovecha su gran estabilidad y sus innovaciones tecnológicas, de forma que Molinux ha conseguido ser una de las distribuciones más destacadas en el panorama nacional debido, en buena parte, a sus procesos de desarrollo y controles de calidad. Además el sistema se integra dentro de la iniciativa UDA, marco de trabajo cuyo fin es unificar y compartir ciertos desarrollos de las distribuciones autonómicas.

Otro aspecto importante de la distribución Molinux es su entorno de escritorio totalmente adaptado al idioma castellano, y su facilidad de uso para la gran mayoría de los usuarios. Por otro lado, el sistema de instalación de Molinux ha evolucionado en las últimas versiones hasta convertirse en un sistema de una gran facilidad de uso, de forma que prácticamente cualquier usuario es capaz de instalar Molinux en su equipo sin necesidad de poseer grandes conocimientos técnicos.



Figura 1. Página principal del proyecto Molinux

Cabe destacar también la capacidad de conectividad que ofrece Molinux, para acceder a Internet desde cualquier sitio, mediante las herramientas de conexión con las plataformas de telefonía móvil como son Vodafone y Movistar.

Otra de las aportaciones tecnológicas de Molinux es el *sistema de rescate*, que permite recuperar el sistema en caso de producirse un fallo importante (ya sea hardware, de instalación o sistema) en el menor tiempo posible y sin pérdida de información en la mayoría de los casos.

El sistema también dispone de un completo manual de usuario, disponible en varios formatos desde el sitio web *www.molinux.info*, que cubre todos los aspectos de la instalación y el funcionamiento del sistema y las aplicaciones que lo componen y que complementa a la formación on-line gratuita que sobre este paquete se ofrece a través de la plataforma *http:// formados.ceslcam.com*

Molinux ofrece un sistema de soporte técnico para todos sus usuarios a través de los foros (*http://foros.molinux.info*) y la base de datos de errores (*http://bugzilla.molinux.info*) con el fin de facilitar la experiencia de todos los usuarios. También pone a disposición de éstos la linea telefónica 902-888-335, mediante la cual resuelve las dudas e incidencias que se planteen y un contacto directo por correo electrónico (*ceslcam@ceslcam.com*).

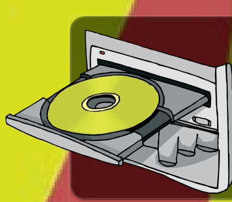
Además, Molinux quiere fomentar la implicación de los desarrolladores y empresas TIC en el desarrollo, para ello facilita el acceso a la forja de desarrollo (*http://forja.molinux.info*) para que todos los interesados puedan seguir la evolución de los proyectos, e incluso plantear la inclusión de sus propios proyectos.

El Gobierno de Castilla-La Mancha, por medio del CESLCAM, tiene en Molinux su principal proyecto dentro de su plan de difusión y promoción de software libre en la región castellano-manchega. y complementando otras líneas de trabajo con estas tecnologías referidas a formación, certificaciones y observatorio tecnológico.



Figura 2. Página de la comunidad de Molinux





Si no puedes leer el disco DVD y no notas ningún tipo de deterioro mecánico, por favor, pruébalo en al menos dos unidades de disco diferentes.

En caso de problemas, escribir a: *dvd@lpmagazine.org*

FreeBSD 7.1

FreeBSD es, para mi, el mejor Sistema Operativo que existe. En tres palabras: sencillo, estable y seguro. Durante largo tiempo ha sido una de mis mejores experiencias en sistemas tipo Unix. En resumen... siempre he encontrado una aplicación para lo que he necesitado. (Anon)

Esta es la descripción de un usuario experimentado en FreeBSD. Tenga algo en cuenta, la distribución que hoy le mostramos, es más que un regalo... es una garantía para su seguridad.

Lo nuevo

Publicado el 5 de enero del 2009, con nosotros uno de los mejores sabores de la gama BSD. Ha llegado con una nueva versión, esta vez la 7.1, con el objetivo de ofrecerle lo mejor de si mismo.

FreeBSD, un sistema operativo libre basado en BSD (Berkeley Software Distribution) que incluye software del proyecto GNU, viene armado con lo último en gestores de escritorio, Gnome en su versión 2.22.3 y KDE en la rama 3.5.10. Con ello dispone todo sobre sus manos para unir en un solo elemento comodidad y protección.

Incluye ahora capacidad de *journaling* en su sistema de archivos UFS. Una característica que provee de un control de transacciones sobre los ficheros evitando una posible corrupción de su sistema.

Destaca también por su innovador *ULE scheduler*, incluido por defecto en los kernels desarrollados para procesadores *i386* y *amd64*, que junto con el desarrollo del nuevo algoritmo para gestión de memoria *jemalloc*, hace las delicias de aquellos usuarios que disponen de procesadores con múltiples núcleos o aquellos otros que dedican sus esfuerzos a desarrollos de computación paralela.

Requisitos

FreeBSD soporta múltiples arquitecturas de procesadores, y como muestra de sus escasas exigencias, aquí tiene los requisitos solicitados para un PC normal con un procesador de la casa Intel:

- Procesador 486 o superior.
- Un mínimo de 24 MB de RAM.
- 150 MB de espacio libre en disco (instalación mínima).

Instalar FreeBSD sigue siendo bastante sencillo, sobre todo si alguna vez ha probado a configurar una distribución como Linux Debian en su sistema. Para que vea que esto es cierto, puede seguir el proceso de instalación descrito en el número 30 de esta misma revista y comprobar que los pasos serán idénticos. Puede encontrar la revista online en la sección de descargas de nuestra página web.



Migración

Si usted es un recién llegado del Sistema Operativo Linux, hay 3 elementos que pueden salvarle la vida aún tras una instalación exitosa por su parte.

Por norma general estará acostumbrado a utilizar un entorno gráfico como Gnome o KDE y tal vez haya escogido los paquetes correspondientes durante la etapa de instalación. Después de haberse registrado por primera vez en su sistema y arrancar el entorno gráfico con el comando `startx`, podría no ver iniciado el entorno gráfico elegido. Esto puede resultar frustrante, pero tiene fácil solución. Debe editar o crear un archivo `.xinitrc` en su directorio `$HOME` con el siguiente contenido:

- Para usuarios de KDE: `/usr/local/kde4/bin/startkde`
- Para usuarios de Gnome: `/usr/local/bin/gnome-session`

Si por alguna razón el ratón no funcionase correctamente, edite con permisos de administrador (`root`) el archivo `/etc/rc.conf` y añada las siguientes líneas:

```
moused_type="auto"
moused_enable="YES"
```

Una vez Gnome esté iniciado, si su teclado no responde correctamente a las pulsaciones, puede cambiar su configuración en el menú *System->Preferences->Keyboard*, en la pestaña *Layouts* tal vez tenga configurado por defecto el idioma de EE.UU. Agregue *Spain*, establézcalo por defecto y elimine el anterior.

Seguramente usted no tenga que realizar todos o ninguno de estos pasos, pero como puede observar, son consejos que le podrían ayudar en alguna ocasión.

Gestión de paquetes/ports

En FreeBSD es muy común escuchar el término *port* a la hora de tratar con software. Debe entender primero la diferencia existente entre la denominación de *paquete* y la de *port*. Los paquetes son precompilados que están listos para su instalación, en linux son más conocidos como `*.deb`, `*.rpm`, etc...En cambio, podríamos definir los *ports* como el código fuente sin compilar de un programa, que normalmente viene empaquetado en un `tarball` y quizás comprimido mediante *Gzip* o *Bzip*,

Tabla 1. Multimedia en FreeBSD 7.1

AUDIO	VIDEO	TV / RADIO
XMMS	Mplayer	XawTV
Cdda2wav	Xine	FxTV
Lame	Totem	WmTune

y que están preparados para ser compilados en su sistema local y así crear un ejecutable ligero adaptado a tu sistema.

Pensará que es más complicado, pero se equivoca. Al estilo de `apt-get` en debian, les presento la herramienta de gestión de ports de FreeBSD, `pkg_add`. Será utilizada directamente desde una consola de comandos. Sus dos usos principales son los siguientes:

- Si ha descargado un port desde un FTP oficial: `$ pkg_add nombreport-v.x.y.tgz`
- Si desea buscarlo directamente en los FTP: `$ pkg_add -r nombreport`

Como puede observar, la opción `-r` le evitará molestias y hará el trabajo sucio por usted, descargará el port, lo compilará y situará los binarios en el lugar adecuado. Si desea ver información sobre sus instalaciones, encontrará todo lo necesario en el directorio `/var/db/pkg`.

Desarrollo

No existe mejor situación que un Sistema Operativo robusto para el desarrollo de software robusto. Tanto si es un aficionado a la programación como un usuario experimentado, dispondrá de todas las herramientas necesarias para escribir su código, compilarlo, y depurarlo.

La última versión de la rama GCC4 viene entre las novedades más destacadas de esta distribución. Si no es un amante del C, no se preocupe, FreeBSD le ofrece al mismo tiempo soporte, como una infinidad de módulos y librerías adicionales para lenguajes como:

- Perl,
- Python,
- Ruby,
- PHP,
- Java.



Kdevelop viene con fuerza para aquellos otros que precisen de un IDE con todo el material de desarrollo integrado en una misma aplicación.

Seguridad

Sin duda alguna, seguridad es el punto fuerte de este Sistema Operativo. Fue diseñado con esa idea en mente y así ha continuado durante largos años. Veamos algunas de sus principales características:

- Algoritmos de cifrado: DES, Crypt, MD5 (por defecto).
- TCP Wrappers para permitir o denegar acceso a sus servicios.
- Autenticación con Kerberos.
- Capa de transporte cifrado OpenSSL.
- Administración remota con OpenSSH.
- IPSec sobre Redes Privadas Virtuales.
- Listas de acceso y control de archivos.
- Uso de *jaulas* (conocidas en linux como `chroot`).
- Y mucho más...

Quizás uno de los puntos más importantes sea su sistema `portaudit`. Su objetivo principal es consultar una base de datos actualizada y mantenida por profesiones en seguridad y por los propios desarrolladores de FreeBSD para encontrar las últimas vulnerabilidades descubiertas en su software.

Tan sólo precisa de un comando para mantenerse al tanto de los problemas:

```
# portaudit -Fa
```

El parámetro `-F` actualiza automáticamente la base de datos en `/var/db/portaudit`. Con `-a` comprobará si alguno de los ports que tiene instalado padece algún problema de seguridad. Es cierto, nunca antes administrar un sistema de forma básica resultó tan sencillo.

Compatibilidad

En FreeBSD existe un término llamado *Compatibilidad Binaria con Linux*. Es de especial interés puesto que es una característica que le permite ejecutar en su sistema el 99% del software desarrollado para Linux. No ocurre lo mismo en la situación inversa y esto es un punto a favor para FreeBSD.

Probablemente haya elegido esta opción entre sus paquetes en la etapa de instalación. De todos modos siempre puede activarlo mediante dos simples comandos:

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

Recuerde que estas tareas de administración deben ser ejecutadas con permisos de superusuario `root`.

Multimedia

Alguno puede estar pensando que FreeBSD es un sistema dedicado solo a servidores, o entornos aplicados a I+D+I, si bien ésta es una de sus funcionalidades, como ningún otro también está orientado a entornos de escritorio normal. Tiene soporte para una infinidad de tarjetas gráficas y de sonido así como a toda clase de dispositivos Plug&Play.

Entre las aplicaciones que más llaman la atención podemos destacar las que están mostradas en la Tabla 1.

Y esto es nada más que una pequeña muestra de la ingente cantidad de paquetes que encontrará en el interior de su DVD particular.

Conclusión

Qué más se puede pedir. Con usted la posibilidad de disponer de un sistema prácticamente *blindado* por defecto y que le ofrece una versatilidad inigualable. Aprovechemos ahora que los tiempos han cambiado y todavía existen muestras de generosidad. Lo que hoy mostramos es el resultado del esfuerzo de toda una comunidad dedicada al software libre. Por usted y para usted.

Aprenda, disfrute, y haga de FreeBSD su nuevo compañero, recuerde que Beasty siempre le acompañará.

David Puente Castro, alias *blackngel*

Versión inestable del sistema de ficheros Btrfs

La versión en desarrollo de Linux (2.6.29) ha incluido una versión inestable del sistema de ficheros Btrfs para propósitos de desarrollo, no de producción. Btrfs es un sistema de ficheros copy-on-write creado como respuesta a ZFS. Está siendo desarrollado por Chris Mason de Oracle. Más información: *http://www.reddit.com/r/linux/comments/7oobp/btrfs_merged_for_2629/* y *http://diegocg.blogspot.com/2008/03/progreso-de-btrfs.html*
Fuente: *http://softlibre.barrapunto.com/article.pl?sid=09/01/10/133214*

Vietnam migrará a Software Libre

A mediados de este año Vietnam migrará sus reparticiones públicas a software libre. *El ministerio de Información y Comunicaciones de Vietnam emitió una instrucción sobre el uso de software libre en los organismos estatales que potencia los programas de código abierto en las reparticiones públicas como ministerios, gobernaciones y municipios.*
Según la nueva normativa, para el 30 de junio de este año, el 100 por ciento de los servidores de la administración estatal deben correr este tipo de aplicaciones y el personal de administración debe estar entrenado para su gestión y uso.
El paquete de programas de productividad OpenOffice, el navegador Firefox y el cliente de correo Thunderbird debe ser usado en 70 por ciento por las administraciones a final de 2009. El ministerio de Información y Comunicaciones de Vietnam emitió una instrucción sobre el uso de software libre en los organismos estatales que potencia los programas de código abierto en las reparticiones públicas como ministerios, gobernaciones y municipios, comenta *cooperativa.cl*.
Fuente: *http://www.cooperativa.cl/prontus_nots/site/artic/20090108/pags/20090108110814.html*

El País promueve el uso de GNU/Linux

El 8 de enero de 2009 el diario español El País (*www.elpais.com*) publicó un artículo sobre GNU/Linux en el cual anima a sus lectores a probar este sistema operativo. Entre sus ventajas menciona: bajo coste y aptitud para los ordenadores viejos. Aquí unos fragmentos del artículo: *Además de los posibles sistemas operativos con los que ya haya trabajado, y de los que probablemente tendrá una copia en un CD, DVD o incluso disquetes, siempre puede optar por Linux, el sistema operativo libre y gratuito por excelencia (...) La mayoría de versiones (distribuciones) de Linux no ocupan mucho espacio y los recursos necesarios son mínimos (...) Una vez instalado, se puede acceder a todo tipo de aplicaciones, también gratuitas, para Linux.*
Fuente: *http://www.elpais.com/articulo/semana/Gratis/total/elpeputeccib/20090108elpciblse_4/Tes*

¿Comodidad o seguridad?

Uno de los grandes paradigmas de la seguridad informática es que hacer un sistema seguro lo vuelve incómodo. En un coche, no es cómodo llevar un cinturón de seguridad, en un aeropuerto no es cómodo pasar por un control policial… y en un ordenador, no es cómodo estar poniendo usuarios y contraseñas.

Sin embargo, para poder usar hoy en día un ordenador, deben llevarse a cabo una serie de comportamientos que lo vuelven incómodo. Instalar un cortafuegos y un antivirus, crear diferentes cuentas de usuario, actualizar el sistema y sus aplicaciones, cuidar en qué páginas web navegas…

A pesar de eso, hace poco tuve la oportunidad de encontrarme con un ordenador que parece que no necesita nada de eso. Se trataba de un *ultraportátil*, que con un poco de memoria, un poco de pantalla y un poco de sistema operativo te permite hacer de todo. Incorpora sus conexiones de red para conectarte a Internet, y a partir de ahí, el mundo es tuyo.

Se supone que los usuarios de esos sistemas son básicamente dos, los que se inician en el mundo de la informática y aquellos que necesitan un sistema portátil realmente ligero para sus desplazamientos. Basado en Linux y con aplicaciones gratuitas, el precio es realmente bajo, por lo que no se trata de un dispositivo para altos ejecutivos, como los primeros portátiles (por llamarles así) de principio de los 90.

Su manejo es muy cómodo a pesar de su mini teclado y la pantalla se ve bien aunque sea pequeña. Pero ¿dónde está la seguridad de ese sistema? A primera vista, lo único que se puede hacer es acceder a las aplicaciones y conectarse a Internet, no hay un sistema de configuración de un cortafuegos, ni un antivirus, ni un sistema de gestión de cuentas de usuario.

Es maravilloso. ¿Y su seguridad?, ¿La tiene?, ¿Qué va a hacer un usuario no entrenado cuando tenga que aplicar un parche al sistema?, ¿Va a evitar que sea víctima de un phishing?, ¿Le seguirá llegando spam? No queda muy claro todavía…

Estos sistemas son muy bonitos, muy cómodos de usar…. pero inseguros. En manos de una persona que sepa lo que hacer, puede que no se vuelva una bomba de relojería. Pero los usuarios novatos que creen encontrar un chollo informático por menos de 300 euros lo pueden pasar muy mal.

Por no hablar del ejecutivo que necesita estar a la última y presumir de ordenadorcito en un viaje en avión: la información que esté en ese sistema puede que quede tan expuesta a los intrusos como si la publicara en primera página de un periódico nacional.

La seguridad es incómoda, sí, y el trabajo que estamos haciendo los desarrolladores es muy grande para que sea más cómoda. Pero hacer que un sistema sea sencillo simplemente eliminando seguridad o *escondiéndola* no beneficia a los usuarios, más aún, los pone en grave peligro sin ninguna necesidad.

Pero que conste que el sistema me gustó, es muy bonito, se presume mucho cuando lo sacas en un aeropuerto a la hora de pasar el control de seguridad.

Fernando de la Cuadra, Ontinet

La nueva versión de gnuLinex Lenix

La Junta Extremeña anunció hace más que un mes que está disponible para la descarga desde el 12 de enero la nueva versión de gnuLinex Lenix, que está basada en Debian Lenny. Lenix incluye el Kernel Debian Linux 2.6.26-1-686, el sistema gráfico X.org 7.3 y el escritorio Gnome versión 2.22. Como suite ofimática incluye OpenOffice 3.0 y el navegador Iceweasel (Firefox) versión 3.0.4. Los requerimientos mínimos de Lenix son 4 Gb de disco duro, aunque se recomienda 8 Gb, un procesador Pentium III o AMD K7 a 800 MHz o superior. Tarjeta gráfica Vesa 2.0 o compatible y un monitor con soporte de 1024×768 a 24 bits.

Página web *www.historiasdequeso.es*

Más información:
http://www.historiasdequeso.es/2009/01/gnulinex-05-2-estable-ya-disponible-nombre-en-clave-lenix.html
Fuente:
http://softlibre.barrapunto.com/article.pl?sid=09/01/11/1842201

AXARnet COMUNICACIONES

--- especialistas en registro y alojamiento desde 1996





A tu lado siempre, ofreciéndote lo mejor, al mejor precio.

Servidores Dedicados

- Windows o Linux.
- Servidores Dell de última generación
- Centro de datos en Madrid
- Garantía SLA
- Paneles de control Plesk
- Administración completa**
- Conectividad Multihomed
- Soporte 24x7

Producto Recomendado



Desde 89 €*/mes



** Servicio opcional con coste adicional

Registro de Dominios

Registra tu dominio desde

1,99 €*/año

Incluye 2 buzones de correo, Webmail y Redirección web

Servidores VPS



Desde 6,40 €*

- Recursos garantizados
- Totalmente escalables
- Instale sus propias aplicaciones
- Panel de Control Plesk o CPanel
- Tecnología Virtuozzo
- Acceso root (SSH)

¡Su servidor al precio de un alojamiento compartido!

Prueba 24 horas **GRATIS**







Alojamiento Web

Desde 250MB de espacio en disco.

Tráfico Ilimitado.

Antivirus y Antispam gratuitos



Desde 3,99 €*

* Iva no incluido en los precios





Linux es fácil

20minutos.es anuncia que Linux se ha convertido en una distribución más fácil para el usuario mediocre: *Hace unos años Linux era sinónimo de problemas y dificultad para el usuario común, acostumbrado a la simplicidad de manejo de Windows. Hoy en día, las empresas y organizaciones que promueven este sistema operativo libre se han esforzado por mejorar su entorno gráfico hasta lograr que el cambio desde Windows sea una tarea fácil*. Entre las distribuciones más fáciles de utilizar enumera a Ubuntu, Fedora, OpenSuSE y Mandriva. Hace también un repaso interesante por las distribuciones españolas: *Las administraciones han creado las suyas propias tras el ejemplo pionero y exitoso de gnuLinEx, realizada por Extremadura a partir de Debian. Como su misión consiste en extender el uso de la informática y el acceso a Internet de la población, se caracterizan por su sencillez y por estar bien adaptadas a los usuarios españoles. De esta manera han surgido Guadalinex en Andalucía, Molinux en Castilla-La Mancha, Max en Madrid y Lliurex en Valencia, todas ellas basadas en Ubuntu. LinKat se creó en Cataluña, derivada de openSUSE y en catalán. Por otro lado, hay una distribución de Linux realizada en Galicia, denominada Trisquel, cuyo idioma por defecto es el gallego pero que también ofrece traducciones al español y al inglés, junto al catalán y el euskera. La idea de este proyecto nació en la Universidade de Vigo.*
Fuente: *http://www.20minutos.es/noticia/441140/0/linux/usuario/medio/*

Linux4one

Cansado de Linpus?, Si eres de los que buscan una buena alternativa Linux para la netbook popular del 2008, les presentamos Linux4one. Una distribución Linux basada en Ubuntu y personalizada para la Acer Aspire One. En un principio esta versión solo estaba disponible en italiano pero en su versión RC 2 ya está disponible en Inglés y se esperan futuras versiones con más idiomas (incluyendo el español).
Linux4One RC2 tiene como característica la interfaz de Ubuntu Netbook Remix para pequeñas pantallas y un kernell pulido para soportar el Hardware de la Aspire One, incluyendo la ranura para tarjetas SD, micrófono y WiFi. El sistema operativo está basado en Ubuntu 8.04 (Hardy Heron) que es la última versión de soporte, pero no la última versión de Ubuntu (Intrepid Ibex) como en le caso de la distribución de Easy Peasy para las Eee PC. El Sistema Operativo también viene con un gran número aplicaciones pre cargadas, incluyendo Flash 10, OpenOffice.org 3.0 y el reproductor multimedia VLC. La mayoría de estas aplicaciones están disponibles en los repositorios de Ubuntu, pero es bueno ya tenerlos incluidos. Por supuesto, si prefieres puedes eliminar o sustituirlos usando el Gestor de paquetes Synaptic.
Fuente: *http://www.fayerwayer.com/2009/01/linux4one-una-alternativa-linux-para-tu-aspire-one/*

El nuevo dispositivo de Palm

El nuevo dispositivo de Palm, llamado *Pre*, dispone de una pantalla multitáctil y detecta la posición horizontal/vertical de forma similar al iPhone, pero además tiene oculto bajo la pantalla un teclado QWERTY deslizante que recuerda al de los anteriores modelos *Centro* ó *Treo*. La plataforma webOS está basada en Linux y de alguna forma es una extensión de lo que habían estado preparando durante algún tiempo, pero sobre el núcleo de Linux hay una capa de presentación que consiste básicamente en WebKit: todas las aplicaciones, incluyendo las propias de Palm, estan hechas con HTML, CSS y Javascript, y pueden hacer llamadas a partes más profundas del S.O. para ciertas funcionalidades. Esto significa que casi cualquier desarrollador web puede programar una aplicación para este smartphone. Palm ha confirmado que habrá una tienda online de aplicaciones para el nuevo sistema y SDK disponible públicamente.

Fuente: *http://softlibre.barrapunto.com/article.pl?sid=09/01/09/152232*

¿El Gobierno quiere promover el software libre en la Administración?

Según la revista *Expansión.com* el Gobierno quiere promover el software libre en la Administración. Como dicen, *el Gobierno quiere aprovechar la nueva ley de acceso electrónico de los ciudadanos a los servicios públicos para impulsar en las administraciones públicas una amplia adopción del software de código abierto (o software libre) frente a soluciones propietarias (de pago por licencias).*

Una iniciativa que perjudica directamente al negocio de Microsoft, cuyos programas cuentan hoy con una destacada presencia en algunas comunidades autónomas españolas –sobre todo, en aquellas regiones consideradas más ricas o industrializadas.

La ley, que entrará en vigor en enero de 2010, reconoce el derecho de los ciudadanos a relacionarse con la Administración por medios electrónicos.

Además, regula los aspectos básicos de la utilización de las tecnologías de la información en la actividad administrativa, en las relaciones entre las distintas administraciones públicas, y en las comunicaciones entre organismos y ciudadanos.

A efectos prácticos, las instituciones deberán colgar todos sus servicios en la red para que sean accesibles a los ciudadanos. Esto implica que muchos organismos deben acometer una importante labor de modernización tecnológica, que abre la puerta al empleo del software libre.

Fuente: *http://www.expansion.com/2009/01/07/funcion-publica/1231318890.html*

Linux se hace más popular gracias a los netbooks

El portal revistadeinternet.com confirma lo que los autores de Linux+ predican desde hace más que un año: los netbooks extienden el uso de Linux. Isabel Fuentes escribe en el artículo publicado el 7 de enero que *el mercado de estos pequeños portátiles ha adoptado Linux mucho más que los PC o los notebooks. Mientras que los netbooks que funcionan con Linux no han conseguido demasiada aceptación entre los usuarios de Estados Unidos, su éxito es mucho mayor en otros países, sobre todo en los que están en vías de desarrollo. Gownder indica que a pesar de que la mayoría de los netbooks utilicen Windows, el éxito de Linux por la gran penetración en el mercado que ha tenido*. Añade que *la demanda de netbooks es grande, ya que un tercio de los hogares de Estados Unidos piensa comprarse uno, dado que se consideran como un segundo o tercer ordenador, especialmente usado por niños o por personas que necesitan moverse continuamente. Las buenas noticias para las empresas fabricantes es que parece que este nuevo segmento del mercado no va a invadir el de los portátiles tradicionales, ya que sólo una cuarta parte de las personas a las que se ha consultado para realizar este informe afirma que le gustaría conseguir un netbook para reemplazar a su portátil.*

Fuentes: *http://www.revistadeinternet.com/Comunicaciones/1988/2009/01/07/Los-netbooks-extienden-el-uso-de-Linux* y *http://www.vnunet.com/vnunet/news/2233381/netbooks-sales-driving-linux*



Profesional

Soluciones integrales de alojamiento

Completo

Una gama completa de productos y servicios de presencia en internet: dominios, alojamiento compartido, servidores dedicados, SaaS...

Flexible

Soluciones escalables según sus necesidades.
Disponibilidad inmediata.
Sin gastos de alta.

Fiable

Alojamiento en datacenter de última generación.
Alto nivel de rendimiento, disponibilidad y estabilidad garantizados.

Seguro

Asistencia técnica avanzada y personalizada.
La garantía de uno de los líderes europeos en servicios de Internet.

Hosting/Dominio
Pack todo incluido desde 1€/mes.

E-Commerce
Su tienda online desde 11,99€/mes.

Servidores Dedicados
Máxima potencia desde 49€/mes.

A precios sin competencia



Amen es uno de los principales proveedores de servicios de Internet de Europa, presente en España desde 2001 y con más de 15.000 clientes españoles que confían en nosotros.

Si cree en Internet, creerá en Amen.

902 888 411
www.amen.es

Ubuntu Eee ahora se llama Easy Peasy

En números anteriores, hemos hablado de Ubuntu Eee, una versión de la distribución especialmente orientada al netbook de Asus, Asus Eee PC, el modelo que inauguró (o más bien inventó) este nuevo mercado, el de los netbook.
El cambio de nombre viene motivado por el eslogan del Asus Eee: Easy to learn, work, play; Excellent Internet experience and Excellent mobile computing experience, y de manera similar los creadores de Ubuntu Eee han decidido nombrarlo: Easy Peasy.
Este cambio de nombre está presente desde Enero de 2009, que fue la fecha en que se lanzó la versión de Ubuntu Eee (o Easy Peasy), basada en la versión 8.10 de su distribución matriz, Ubuntu 8.10 Intrepid Ibex.

Galinux, el Linux gallego

Siguiendo la estela que comenzó la Junta de Extremadura con gnuLinEx y que han seguido muchas otras comunidades autónomas, Galicia acaba de apuntarse al *carro* de las comunidades con distribución propia basada en GNU/Linux y lo ha hecho con una recién creada distribución llamada Galinux, que está basada en Ubuntu.
La Xunta de Galicia ha colgado un vídeo de demostración y ha proporcionado 5.000 ejemplares de manera gratuita para que los gallegos que quieran la prueben. Hay un vídeo promocional de la misma disponible en la red, con muy buena calidad y muy interesante de ver.
En la red:
http://www.galinux.org

¿Wine en los repositorios de Ubuntu?

Scott Ritchie, uno de las personas más influyentes en Ubuntu, de hecho es un MOTU (Master of the Universe de Ubuntu), propuso hace unos días en la lista de correo de desarrolladores, que Wine, esté incluído en el repositorio *main* (principal) de Ubuntu, lo que prácticamente lo incluiría por defecto en cualquier instalación de esa distribución.
El objetivo siguiente es que cuando alguien haga click en un archivo ejecutable de Windows, una ventana emergente informe sobre las posibles acciones y posteriormente pueda configurarla y ejecutarla con Wine. De todas maneras, para llegar al objetivo que os comentamos todavía queda camino por andar, porque habría que modificar también parte de Gnome y otro software.

Canonical Store, otra forma de regalar

Pasadas ya las fechas navideñas, probablemente más de uno haya tenido problemas en regalar, porque ya sea por precio o por falta de ideas muchas veces no sabemos qué comprar para regalar a las personas más cercanas. Posiblemente, también hayáis tenido problemas en decirle a los que os regalaban qué queríais. Este texto está más orientado a este segundo caso, el decir qué queréis regalar cuándo os preguntan. Ya que habrán pasado las Navidades cuando lo leáis, espero que os resulte útil para años posteriores, o para cumpleaños, santos o cualquier otra ocasión en la que regaléis u os regalen. Muchos ya conoceréis alguna tienda de regalos en línea de esta u otras distribuciones o fundaciones, si queréis una lista más amplia, podéis consultar al final de este artículo y encontraréis una lista con los sitios web de algunos más que os pueden interesar.

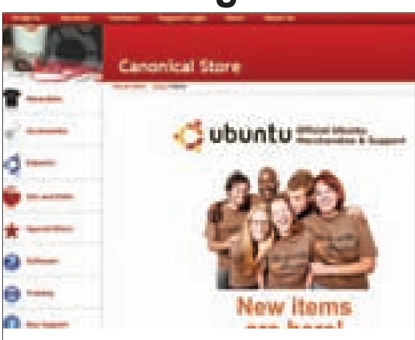
La idea de estas tiendas, es ofrecer a los aficionados de Ubuntu, Kubuntu, Xubuntu... en este caso o de la distribución o software que corresponda en cada caso, todo tipo de artículos realmente curiosos con el logo o algún rasgo identificativo de los mismos. Algo muy parecido a lo que ocurre con los artículos oficiales de los clubes de fútbol.

Además de suponer una forma de financiación, es una manera más de difundir la marca y por lo tanto de hacer marketing y ganar algo más de conocimiento entre el gran público. Sin lugar a dudas, además de suponer un regalo original o un artículo muy curioso que comprar, cuando regalamos o compramos un artículo de estas tiendas, estamos ayudando a soportar a tantos desarrolladores que de manera más o menos desinteresada nos ofrecen software de tanta calidad como por ejemplo en el caso de Canonical Store, Ubuntu.

Y es que a pesar de las críticas que puedan levantarse en los sectores más radicales



Galinux



Página web de Canonical Store

dentro del Software Libre, la financiación es algo básico para todos, y ya que tenemos un modelo de negocio dentro del Software Libre que es menos propicio para la venta de software en sí, sí que tenemos que aprovechar este tipo de modelo de negocio, totalmente justo y en línea con los ideales del Software Libre. Por ahora no supone un porcentaje alto de ingresos para las empresas dedicadas a este mundo, pero sería ideal que así lo fuera, como lo es para los grandes clubes de fútbol y otros deportes y que sea un apoyo más en este campo tan necesario, el financiero.

Como todos sabréis, el soporte y la venta del software *como servicio*, que en parte es lo mismo, son las bases de una *industria*, la del Software Libre, que con muchos menos recursos que la industria del software comercial, nos ofrece unos productos de mucha mayor calidad en muchos casos y con un coste cero. De hecho, hasta te regalan y te envían gratuitamente discos compactos (CDs) con dicho software para que los pruebes y lo instales gratuitamente con programas como el ShipIt de Canonical o algún otro que están imitándolo.

Esperemos que la tendencia sea la del crecimiento y que poco a poco ésta sea otra fuente de financiación para el Software Libre y además nos ayude a todos a elegir un regalo útil y curioso, además de convertirse en todo una *envidia* para nuestros amigos que también sean aficionados a este mundo.

En la red (en inglés):

- Canonical Store: *https://shop.canonical.com/*
- Mozilla Store: *http://intlstore.mozilla.org/*
- Mandriva Store: *http://store.mandriva.com/*
- The openSUSE Shop: *http://shop.opensuse.org/shop-eu.html*



Gazelle Ultra

Hace unos años era impensable que hubiera portátiles o equipos de escritorio que trajeran preinstalado con una licencia OEM, Ubuntu o cualquier otra distribución de GNU/Linux, pero Dell y otras compañías dieron el paso, porque sabían que atrás, había un público esperando. Este tipo de movimientos no se producen en las empresas hasta que saben que van a tener una aceptación, y dado el progresivo aumento de usuarios de Ubuntu en concreto y de Linux en general, se producen cada vez con mayor frecuencia. Pasó el tiempo y poco a poco, GNU/Linux se ha convertido también en el sistema operativo estrella de los nuevos netbooks, portátiles de bajo coste y reducido tamaño. Como podéis observar, de ser prácticamente un desconocido a estar presente en mucho de los centros comerciales, hay un gran paso y estos casos nos dan esperanzas de que poco a poco van a ir apareciendo más y más dispositivos con GNU/Linux y que en un futuro a medio plazo, ya no digo a largo como decía antes, Linux sea una alternativa real a Microsoft Windows en el mercado de escritorio.

Dejando de dar rodeos alrededor de la noticia que nos ocupa, vamos a ir al centro de la misma, al modelo de portátil llamado *Gazelle Ultra*. Es un producto de System76 y entre sus características podemos destacar: 13.3" de pantalla, lector biométrico, CPU Core Duo T3200 2.0 GHz, 2 Gb de RAM DDR2 de 800 MHz, video Intel X4500HD, disco de 160 Gb, regrabadora de DVD, Wi-Fi Intel Link 5100 y un año de soporte técnico.

Después de leer todo esto os estaréis preguntando por qué os anuncio este modelo en una sección de Ubuntu. Como podréis ver en la imagen que os dejo en esta misma página, el portátil está decorado por un enorme logo de Ubuntu y además trae preinstalado Ubuntu en su versión de 64 bits. Aparte de ésto, que es ya llamativo por el logo, tenemos que destacaros que este modelo en concreto, sólo se vende con Ubuntu Linux.

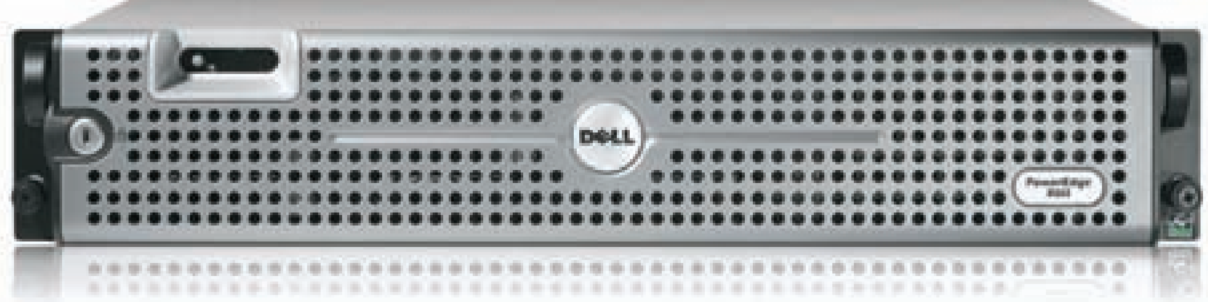
Y este sólo se vende con Ubuntu Linux, es lo que queremos destacar y celebrar, que existen portátiles y equipos que traen exclusivamente Ubuntu y lo usan como un *arma* de marketing. Sin lugar a dudas, la tendencia es más que buena para nosotros y en menos tiempo del esperado, todo podremos estar viendo en breve, quien sabe, si versiones de juegos como ISS Pro para GNU/Linux. Sólo es cuestión de tiempo (y no demasiado).

Más detalles en:
http://www.workswithu.com/2008/12/12/system76-launches-biometric-ubuntu-linux-laptops/



PUBLICIDAD





Hosting Profesional Linux y Windows
Planes de Hosting para revendedores
Servidores Dedicados
VPS (Virtual Private Server)
Planes Duales (Linux+Windows)

www.sys4net.com

No arranca el modo gráfico, sólo el modo texto

DISTRO: Mandriva

Es increíble pero cierto: en algunas ocasiones al instalar Linux (sea cual sea la distro) no arranca el modo gráfico. Hoy trataré de explicar cómo resolver el problema; también sé que para algunos este tema ya está más que dominado. Agradezco su paciencia.

Bueno, como comentario personal, pienso que Linux se ha vuelto difícil para la gente común debido a que muchos de nosotros, los linuxeros, tratamos de hacerlo todo en modo texto. Si bien es cierto que muchas de las configuraciones que se hacen en modo texto tienen su réplica en la interfaz gráfica, también es cierto que es más rápido en modo texto, además, en esta ocasión es necesario utilizar la consola.

Bien, vamos al punto. Cuando no entramos de manera automática al modo gráfico en distros como Mandriva, Ubuntu, Debian, etcétera, puede ser porque tenemos una tarjeta de vídeo que necesita un driver propietario y que no viene con nuestra distro. De tal manera que, aunque el archivo diga que se está utilizando, por ejemplo, el driver nv para mi tarjeta nVIDIA, es posible que dicho driver no funcione, por lo que no hay modo gráfico.

Buscando el archivo

De entrada, necesitamos registrarnos como superusuario, lo que es igual a acceder al ordenador como el usuario root.

Una vez ahí, debemos buscar el archivo que configura el servidor gráfico: xorg.conf que generalmente se encuentra en el directorio X11, dentro de /etc. Entonces escribimos:

```
# cd /etc/X11
# vi xorg.conf
```

Yo utilizo el editor vi porque me gusta más, pero ustedes pueden usar el que mejor les parezca.

Una vez que estamos editando el archivo, pulsamos la letra "i", sin comillas (solo para el editor vi). Eso nos permitirá introducir texto en el archivo.

En este archivo debemos buscar primero la sección en donde está el driver: Device.

Ahí buscamos la entrada driver, que estará seguida por algún nombre de driver. El nombre del driver que (casi) siempre funciona en Linux es vesa, por lo tanto cambiamos el nombre del driver, por ejemplo:

```
driver "nv"
```

lo cambiamos por:

```
driver "vesa"
```

Ajustando la resolución

En el mismo archivo podremos encontrar la sección Screen. En dicha sección se encuentran listadas las posibles resoluciones, si nuestro monitor no soporta dichas resoluciones, debemos ajustarlas.

Una vez que hemos modificado el archivo debemos guardar los cambios y salir del editor.

Para ello:

- Pulsamos la tecla Esc
- Ahora pulsamos la tecla ":" (tecla dos puntos)
- Pulsamos la tecla "w" para guardar (en minúscula)
- Pulsamos la tecla "q" para salir (en minúscula)
- Pulsamos Enter

Probar

Ahora solamente nos resta:

- Reiniciar el ordenador o
- Escribir el comando startx

El modo gráfico debería mostrarse.

Instalando un driver correcto

Una vez que ya hemos logrado iniciar con sesión gráfica, podemos ir a la aplicación que controla el hardware, por ejemplo en Centro de Control Mandriva, e instalar el driver apropiado.

Al tener un driver apropiado, podemos mejorar la resolución del monitor y la profundidad de colores, así como tener la configuración de 3D.

Problema solucionado.

Listado 1. Un ejemplo de archivo xorg.conf

```
Section "Device"
    Identifier "device1"
    VendorName "nVidia Corporation"
    BoardName "NVIDIA GeForce FX to GeForce 8800"
    Driver "nvidia"
    Option "DPMS"
    Option "TwinViewOrientation" "Clone"
    Option "AddARGBGLXVisuals"
    Option "TwinView"
EndSection
Section "Screen"
    Identifier "screen1"
    Device "device1"
    Monitor "monitor1"
    DefaultColorDepth 24
    Subsection "Display"
        Depth 8
        Modes "1280x800"
    EndSubsection
    Subsection "Display"
        Depth 15
        Modes "1280x800"
    EndSubsection
    Subsection "Display"
        Depth 16
        Modes "1280x800"
    EndSubsection
    Subsection "Display"
        Depth 24
        Modes "1280x800"
    EndSubsection
EndSection
```

Instalación de programas (paquetes) desde la consola

DISTRO: Mandriva

De entrada, vuelvo a decir que podemos utilizar algún programa con interfaz gráfica para instalar software, pero, a petición de uno de nuestros lectores, escribo cómo instalar software desde la consola, o modo texto.

Para la instalación de software desde la consola contamos con la aplicación urpmi.

¿Qué es urpmi?

La aplicación urpmi es para Mandriva lo que apt-get para las distros basadas en Debian, como Ubuntu.

urpmi es una herramienta de administración de paquetes que se encarga de instalar, eliminar, actualizar y hacer preguntas a los paquetes de software de aplicaciones locales o remotas (vía red). Envuelve al administrador de paquetes RPM (RedHat Package Manager) extendiendo las funcionalidades que otorga. Uno de sus principales fines es el de facilitar la instalación de programas con herramientas que desprenden los problemas comunes sobre ausencia de dependencias. URPMI trabaja con las fuentes oficiales de Mandriva Linux o con fuentes externas tales como las que otorga Penguin Liberation Front y otros repositorios no oficiales (información de Wikipedia).

No cuesta nada de trabajo acostumbrarse a utilizar esta aplicación debido a la sencillez de su implementación y la robustez de su funcionamiento.

urpmi busca dentro de la lista de nuestros repositorios los programas que quisiéramos instalar, si está en la lista, entonces busca las dependencias, para instalarlas también, de modo que nosotros no nos preocupemos. Fácil ¿no?

Instalar software

Por ejemplo, si quisiéramos instalar OpenOffice.org escribiríamos (registrados como root, claro) algo como:

```
# urpmi OpenOffice.org
```

Aparecerá una lista del software a instalar, tanto del software que queremos, como de sus dependencias.

Así de sencilla es la instalación de software a través de comandos en la consola.

Debo mencionar que aunque en ocasiones no es necesario, recomiendo ampliamente que se esté conectado a Internet, pues recordemos que urpmi busca en la lista de repositorios y si nosotros dimos de alta un repositorio de Internet y no contamos con la conexión, pues simplemente no se podrá instalar.



Algunos comandos útiles:

- urpmi <nombre> Instala el paquete <nombre> con las dependencias vinculadas,
- urpme <nombre> Desinstala el paquete <nombre> con las dependencias vinculadas,
- urpmq <nombre> Busca en la base de datos el paquete llamado <nombre>,
- urpmf <archivo> Busca entre los paquetes que componen los repositorios del sistema, archivos o directorios que contengan la palabra <archivo>,
- urpmi.update <nombre> Actualiza la listas de paquetes del repositorio <nombre>.

Si te interesa conocer más: *http://www.mandrivalinux.com/cooker/urpmi.html*

Problema solucionado.



Escríbenos

Estamos a tus órdenes para resolver cualquier problema que tengas con tu LiNUX, no dudes en enviar tus preguntas y sugerencias a la dirección: *ajminero@gmail.com*

Estamos para servir y hacer que LiNUX sea más fácil.

Saludos.

Alejandro Minero

Club Pro



ESPAWEB
www.espaweb.es
Especialistas en planes para Distribuidores sobre plataforma Linux y Windows, marca blanca.
Servidores Dedicados y Housing
Proyectos y Soluciones a Medida
Datacenter propio en Alicante
Soporte técnico altamente cualificado, 24*7
902 380 380

Lenguajes, lenguas y dialectos

Lino García Morales

Case Reas y Ben Fry, en su libro *Processing. A Programming Handbook for Visual Designers and Artists* sugieren *Como los lenguajes humanos, los lenguajes de programación pueden ser agrupados en conjuntos relacionados*.



linux@software.com.pl

Esta conexión podría sugerir una conexión más fuerte, pero no es así. El desarrollo de los lenguajes de programación nada o poco tiene que ver con sus parientes humanos. De hecho, los lenguajes de programación se parecen mucho más entre sí, son muchos menos y, se puede decir sin herrar demasiado, que tienen casi las mismas raíces; a diferencias de las lenguas.

Se estima que, en las casi 200 naciones del globo, se hablan más de 5000 lenguas y 41000 dialectos mientras que existen sólo unos cientos lenguajes de programación diferentes producidos por menos de una cuarta parte de todas las naciones.

Según la Wikipedia un lenguaje de programación es un *conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones, y es utilizado para controlar el comportamiento físico y lógico de una máquina*.

Los ciento y un lenguajes de programación

Mucho han cambiado las cosas desde que IBM lanzó el primer ordenador personal pero, probablemente, el cambio de mayor impacto en todo este tiempo es que, debido al acelerado desarrollo de la electrónica, las comunicaciones y la informática (lo que hoy se da en llamar TIC, Tecnologías de la Información y Comunicaciones), ha puesto en manos de cualquier ciudadano de a pie, el ordenador distribuido más potente del mundo: internet. Cualquier ordenador tiene varios Giga hertzios de frecuencia en su reloj, o varios procesadores pero incluso, cuando ni aún así es suficiente, siempre se puede echar mano a miles, incluso millones, de ordenadores distribuidos por todo el mundo, dispuestos a compartir algún tiempo de proceso y garantizar la potencia de proceso requerida.

En el año 2004, O'Reilly diseñó un poster genial de la historia de los lenguajes de programación, donde muestra 50 años dorados, desde 1954 hasta 2004, en el ámbito de la programación (El poster se puede descargar en *http://oreilly.com/news/graphics/prog_lang_poster.pdf*). Por supuesto, no son todos los que están, ni están todos los que son (según O'Reilly son los lenguajes de programación más importantes); sin embargo, lo más interesante de este poster, es que muestra las derivaciones de los distintos lenguajes para crear otros (sus raíces). Por ejemplo, C#, se basa en C++ y Java2.

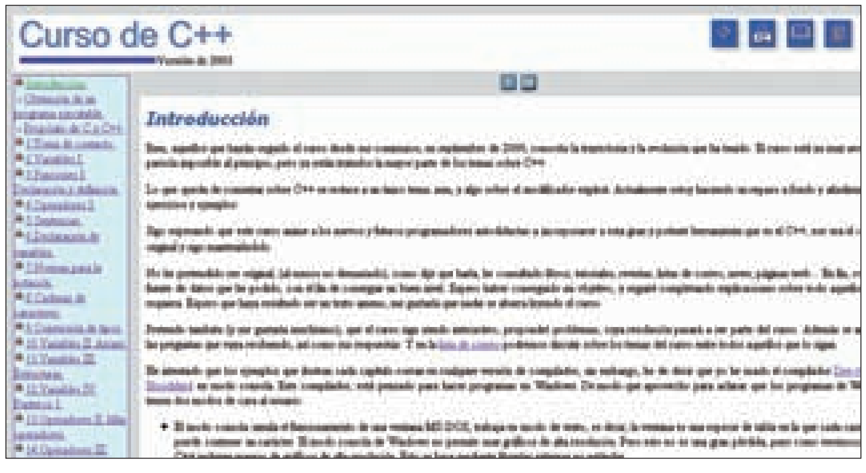


Figura 1. Curso de C++ en castellano *http://c.conclase.net/curso/index.php*

C++, a su vez, se basa en C y Simula que, a su vez, vienen del Algol que proviene de Fortran y Fortran… es el primer lenguaje de programación de alto nivel. Esta evolución evidencia las grandes similitudes y parentescos de la mayoría de los lenguajes entre sí.

Los lenguajes se pueden relacionar en conjuntos según su nivel de abstracción, entorno de desarrollo, forma de ejecución o paradigma de programación.

Nivel de abstracción

El nivel de abstracción está relacionado con la capacidad del lenguaje de ocultar los detalles del ordenador. Los lenguajes de alto nivel están formados por elementos de lenguajes naturales, como el inglés. Sin embargo los lenguajes de bajo nivel interactúan con todas las especificidades del procesador (registros, operaciones, etc.). Las instrucciones del lenguaje de máquina son cadenas binarias (0 y 1) directamente legibles por la máquina (procesador) o completamente incomprensibles (o intratables) para cualquier ser humano (Es importante tener en cuenta que todos los lenguajes operan, finalmente, a nivel de código de máquina. La función última del compilador o intérprete es la conversión a lenguaje de máquina).

Entorno de desarrollo

Los lenguajes pueden ser textuales o visuales. El conjunto de símbolos y reglas sintácticas y semánticas que definen un lenguaje normalmente se pueden expresar textualmente. Sin embargo, los entornos visuales pretenden ayudar a aquellos programadores que piensan espacialmente y prefieren organizar sus pensamientos como relaciones visuales entre elementos.

Forma de ejecución

Los lenguajes pueden ser compilados o interpretados. El compilador es un programa que traduce el código escrito con determinado nivel de abstracción a código de máquina. Sin embargo los lenguajes interpretados no son compilados sino interpretados por otro programa mientras corre. Los lenguajes compilados son más rápidos mientras los programas interpretados pueden ser modificados mientras corren. El refinamiento de tal intérprete ha derivado en máquina virtual. El lenguaje es pseudo compilado a un código intermedio (byte code) que corre en la máquina virtual. Esta técnica facilita la portabilidad del código entre las diferentes arquitecturas o hardware.

Paradigma de programación

El paradigma de programación representa un enfoque particular o filosofía para la construcción del software. Según tal, los lenguajes pueden ser imperativos o procedimentales (es considerado el más común y está representado, por ejemplo, por el C o por BASIC); funcionales (expresivos y matemáticamente elegantes; por ejemplo LISP); lógicos (cuya característica principal es la aplicación de las reglas de la lógica para inferir conclusiones a partir de datos; por ejemplo PROLOG); u orientado a objetos (este paradigma usa objetos y sus interacciones y está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo y encapsulamiento). Un lenguaje completamente orientado a objetos es Smalltalk.

Si bien puede seleccionarse la forma pura de estos paradigmas, a la hora de programar, en la práctica lo habitual es que se mezclen, dando lugar a la programación multiparadigma.

Los pilares del lenguaje

Ante el desafío de la realización de un proyecto una de las preguntas necesarias, tarde o temprano, es, cuál es el lenguaje adecuado. La respuesta no es fácil y depende de muchos factores. Según se baja el nivel de abstracción los programas son más específicos y difíciles de mantener. Depende del grupo de desarrollo condicionará el entorno (y con ello las herramientas de trabajo). ¿Compilado? ¿Interpretado? Los lenguajes interpretados son más adecuados para pruebas rápidas. Sin embargo, cuando se requiera toda la eficiencia, habrá que utilizar algún lenguaje compilado. Por último,



Figura 2. Página del proyecto Perl *http://www.perl.org/*

cuál paradigma será el más adecuado. La respuesta es: depende del problema.

Esto podría conducirnos a alguna conclusión del estilo: Hay tantos lenguajes como problemas; e incluso, si el problema es muy complejo probablemente se necesite del uso de más de un lenguaje para abordarlo. Sin embargo la respuesta no es tan simple y es necesario agregar algunas variables más.

Reusabilidad

Un lenguaje se hace potente en la medida que crece; propiedad que finalmente se puede traducir en un buen soporte (muchas funciones y mucha gente detrás resolviendo problemas). Las funciones se agrupan en librerías o bibliotecas de funciones. Su potencia depende de quién las mantenga y desarrolle (los proyectos de software libre son mantenidos por un mayor número de personas sin presiones comerciales). Su interacción con el lenguaje depende de la tecnología software (incrustada junto con el código, dinámicamente, distribuido, etc.). Desde este punto de vista ¿se podría considerar que C# es un C++ bien soportado?

Disponibilidad

Relacionada con el entorno (arquitectura hardware y software). Los lenguajes, y sus respec tivos compiladores o intérpretes, están disponibles para determinados entornos con, posiblemente, ciertas restricciones de uso. Aún cuando, presuntamente, el lenguaje más adecuado para nuestro problema sea X, la indisponibilidad de las herramientas de desarrollo para el entorno seleccionado será motivo, más que suficiente, de rechazo. La disponibilidad se puede ver igualmente al revés. Suponga que el entorno de trabajo es un navegador web; la aplicación será utilizada vía HTTP. El lenguaje de desarrollo estará limitado por los lenguajes implementados por el navegador (y lo utilizo en modo genérico porque unos a otros convergen en la medida en que se hacen compatibles). De nada vale que sea más adecuado programarlo en LISP si luego no encontramos ningún navegador que permita incorporar un intérprete LISP o, lo que es más grave, si el navegador que utiliza la mayoría de los clientes, no lo tiene.

Durabilidad

El propio poster de O'Reilly muestra el ciclo de vida completo de los lenguajes que referencia (al menos hasta el 2004). Es fácil ver cómo muchos se convierten en otros y mueren. Qué pasa si la aplicación que queremos desarrollar simplemente ponga a prueba el paso del tiempo. Qué pasa con aquellas aplicaciones que duran. Cómo elegir un lenguaje duradero. Por muchos lenguajes que existan pocos han resistido al paso del tiempo. Un buen ejemplo de ello es el lenguaje C. De hecho, se podría hacer en C absolutamente todo lo que en C++ con muy poco esfuerzo. ¿Qué es lo que ha hecho a C tan resistente? Probablemente la conjunción de un buen diseño del lenguaje, la gran disponibilidad de herramientas y librerías y una buena desconexión con la entrada/salida.

Pocos son más

Un buen amigo dice que los lenguajes son necesarios para garantizar la variedad (también para dar de comer a los informáticos) y probablemente tenga algo de razón; pero los lenguajes no son lenguas, ni razas, ni sociedad, sino instrumentos para programar los procesadores de propósitos generales y garantizar determinado funcionamiento. Sacaremos mayor rendimiento a un lenguaje con un dominio perfecto de su sintaxis, reutilizando códigos y fortaleciéndolo.

Supongamos por un momento que disponemos de las herramientas necesarias, para cualquier entorno; para elegir libremente el lenguaje adecuado. Suponga que podemos dividir en tres el nivel de abstracción de las aplicaciones; algo así como bajo, medio y alto. El nivel bajo requiere de un control absoluto del hardware, traducible en código máquina (lenguaje ensamblador) o C, que nos permitirá hacer el mismo trabajo (incluso incluir partes críticas del código en lenguaje ensamblador) con un nivel de abstracción mayor. Para nivel medio el C sería también un excelente candidato, incluso si se quiere emplear el paradigma de programación orientado a objetos pero C no es interpretado. Para muy alto nivel probablemente sería recomendable LISP, con cierto solapamiento con C.

LISP es un lenguaje funcional (utiliza las propiedades matemáticas de las funciones) y multiparadigma, de alto nivel; simbólico (utiliza el símbolo como unidad fundamental) de procesamiento de listas (LISt Procesing);

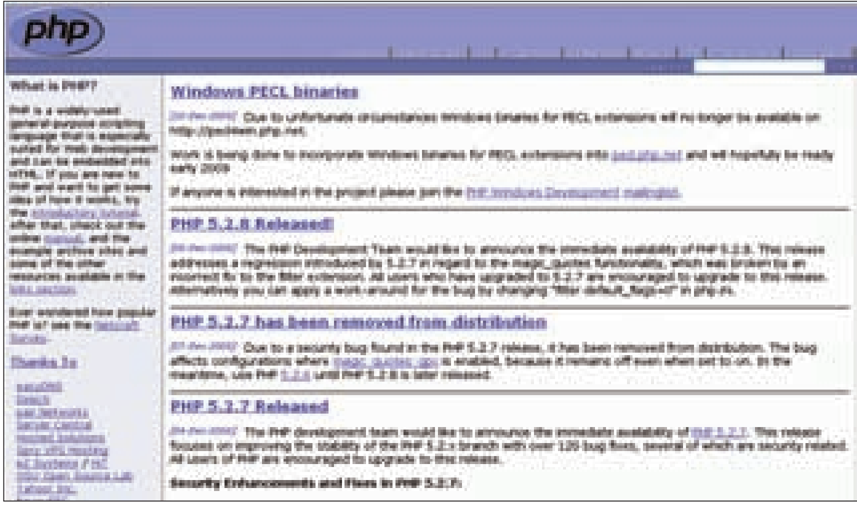


Figura 3. Página del proyecto PHP *http://www.php.net/*



Figura 4. Página del proyecto SmallTalk *http://www.smalltalk.org/main/*



Figura 6. Curso de C

muy flexible (utiliza la misma estructura para datos y código; una de las características de LISP es la posibilidad de tratar las propias funciones como datos (50 años después resulta que es posible considerar XML, tan de moda para el intercambio de datos entre aplicaciones y en arquitecturas orientadas a servicios, como una versión LISP redundante de definición de datos y con alguna alteración en la sintaxis; sustituyendo los paréntesis () por menor/mayor <>) y paso de funciones como parámetro: reusabilidad); autodefininible (entorno a la medida), con un mecanismo de manejo de memoria automático (que libera el espacio utilizado por los objetos que dejan de ser necesitados, lo que se conoce habitualmente como garbage collector); no tiene tipos estáticos, sino que asocia los tipos a los valores en vez que a las variables; eminentemente recursivo (definiciones muy claras y concisas); interpretado (para el desarrollo rápido de prototipos) o compilado y compacto: funcional + recursivo + listas.

Inclusive la arquitectura de LISP (implementado normalmente en C) permite verlo como una máquina virtual susceptible de correr en entornos de bajo nivel de abstracción. Como hemos supuesto una buena reusabilidad se da por hecho un buen soporte para todo tipo de aplicaciones: web, SOA, imagen y sonido, compiladores, procesadores de texto, entornos multiplataforma, matemáticas, etc. y, lo que es más importante, un buen soporte de la comunidad de software libre.



Figura 5. Introducción a XML

Concluyendo

La diversificación de los leguajes de programación es necesaria en la medida que introduzca nuevos paradigmas, debilite la dependencia con los sistemas operativos (potencie multiprocesamiento, distribución y con ello la generalización de entornos de operación), y otras muchas razones. No así la transmisión de sintaxis, o la imposición de determinados usos o tecnologías por razones comerciales; con la excusa de facilitar el trabajo del programador, etcétera. Los ejemplos utilizados en este artículo pretenden demostrar que muchos de los lenguajes más antiguos (como LISP y C) siguen siendo igual de potentes y útiles, y poco o nada tienen que envidiar a los últimos de moda; sin embargo la diversificación de los entornos o herramientas no es tan considerada como la de los lenguajes. Si considerásemos los lenguajes como criaturas y los entornos… vitales, es fácil establecer la conexión. Desde ese punto de vista, potenciando el entorno, probablemente en el poster de O'Reilly, 50 años después, las flechas serían más largas y menos conectadas entre sí. Sin embargo la naturaleza sigue su curso. ¿Cuál será la próxima bestia?

Sobre el autor

Graduado de Ingeniería en Control Automático, Máster en Sistemas y Redes de Comunicaciones y Doctor por la Universidad Politécnica de Madrid. Ha sido profesor en Instituto Superior de Arte, Universidad Pontifici.a *Comillas* y la Universidad Meléndez Pelayo.

Actualmente Profesor de la Escuela Superior Politécnica de la Universidad Europea de Madrid y Director del Máster Oficial en Acústica Arquitectónica y Medioambiental. Lidera grupo de investigación transdisciplinar en la intersección Arte, Ciencia, Tecnología y Sociedad. Becas por la Agencia Española de Cooperación Internacional, FUNDESCO, Consejo Superior de Investigaciones Científicas y Universidad Politécnica de Madrid.

Además ha formado parte de las bandas Cartón Tabla y Música d' Repuesto. Discografía: Las palabras vuelven, Se fue, Bags, Flags, Faqs, Fotos d' parque, These little things that keep inside, Mr. Fro (colaboración con Alejandro Frómeta), El eje del mal, Av abuc, Variaciones en la cuerda VOL. I.

Y ha publicado una novela: *Islas*.

lino@madrid.com

Gambas:
Desarrollo de aplicaciones visuales con Software Libre

Jorge Emanuel Capurro

Gambas es un lenguaje de programación libre, que se distribuye bajo licencia GPL .Provee un entorno de programación ideal para aquellos que buscan desarrollar aplicaciones visuales de manera rápida, potente y sencilla. Gambas es una excelente alternativa de programación libre al legendario Microsoft Visual Basic. Aprendamos un poco sobre este maravilloso lenguaje...



linux@software.com.pl

Hace algunos años, desarrollar aplicaciones bajo Linux era todo un desafío, sobre todo si necesitábamos que éstas implementaran una interfaz gráfica. El desafío consistía en tener que desarrollar la aplicación utilizando diferentes herramientas, todas ellas con un propósito en general, ya que la mayoría son de carácter independiente. Esto quiere decir que, por ejemplo, para poder desarrollar un programa con interfaz gráfica, tuviéramos que desarrollar la *inteligencia* del programa en un lenguaje de programación clásico, como lo puede ser el lenguaje C, y luego, mediante herramientas de diseño de interfaces gráficas, como lo es Glade, integrar ambos trabajos parar obtener el programa final. Esto requiere de diversas herramientas de desarrollo para poder obtener el programa final. Es decir, necesitaríamos un editor de código, un compilador, un depurador, un diseñador de interfaces, como así también, del conocimiento del funcionamiento de cada una de estas herramientas. Todo esto dificultaba mucho a los nuevos programadores que se querían impulsar hacia el desarrollo de software bajo Linux, ya que se necesitaba de varios pasos y del estudio de cada una de estas herramientas para poder desarrollar un producto de carácter profesional.

Para subsanar estos problemas, se crearon los Entornos de Desarrollo Integrado (IDE, Integrated Development Environment), los cuales se encargan de incluir todas, o la mayoría, de estas herramientas en un solo programa. Entre los IDEs para Linux más conocidos se encuentran Kdevelop y MonoDevelop. Kdevelop es utilizado mayormente para el desarrollo de aplicaciones bajo C/C++, mientras que MonoDevelop soporta el desarrollo de aplicaciones .NET, concentrándose principalmente en el lenguaje C#. Pero también existe, y cada día es más popular, Gambas.

Gambas, es un IDE y lenguaje de programación basado en BASIC con extensiones de objetos, que vino al mundo del Software Libre con objetivos de tener una alternativa similar (y más potente) que Microsoft Visual Basic. Gambas es el único entorno en el cual la creación de GUIs (Graphic User Interfaces) está totalmente embebida dentro del código, esto significa que solamente tendremos que *dibujar* los controles en los formularios, y luego codificar las acciones que queremos que éstos realicen. Para decirlo de otra forma, funciona al mejor estilo de Visual Basic. Vale la pena aclarar que, a pesar de que hay muchas similitudes entre ellos, Gambas no intenta ser un clon de Visual Basic, sino mejorarlo, ya que

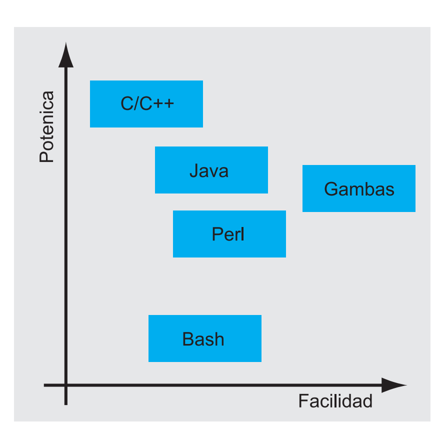


Figura 1. Ubicación de Gambas frente a otros lenguajes de programación

ambos proyectos no son compatibles entre sí. Actualmente un grupo de programadores de Gambas, está desarrollando un componente llamado gb.vb para que en un futuro sean medianamente compatibles, pudiéndose de esta manera lograr que los programadores de Visual Basic que quieran migrar sus aplicaciones a Gambas, puedan hacerlo sin necesidad de tener que reescribir gran cantidad de código. A pesar de no ser compatibles entre sí, los programadores que estén familiarizados con Visual Basic, les será muy intuitivo y sencillo poder empezar a desarrollar sus aplicaciones bajo Gambas. En el gráfico de la Figura 1, podemos ver donde estaría posicionado Gambas frente a otros grandes lenguajes de programación. Como se aprecia, Gambas combina facilidad a la hora de programar, con un buen nivel de potencia como lenguaje.

Elementos principales

Gambas posee distintos elementos que lo conforman, entre los cuales se encuentran:



Significado de Gambas

Gambas, es un acrónimo recursivo (muy populares en los ambientes UNIX), de Gambas Almost Means BASic, que significa *Gambas es casi un Basic*, aludiendo de que si bien Gambas utiliza el lenguaje Basic, no es del todo cierto, ya que se le añade a éste una extensión orientada a objetos. El creador de este lenguaje es Benoit Minisini, que actualmente reside en la ciudad de París.

Según Benoit, a causa de los grandes errores que tiene Visual Basic, decidió crear un lenguaje de programación libre que siga el estilo de Visual Basic, pero que a su vez subsane estos errores y además se pueda ejecutar en plataformas Linux.

- Un Entorno de Desarrollo, el cual brinda las herramientas para el desarrollo de aplicaciones.
- Un Compilador, el cual se encarga de generar a partir de nuestro código fuente, lo que se denomina *Gambas Bytecode*, que es el lenguaje que entiende el intérprete.
- Un Intérprete, encargado de hacer que los programas compilados en Gambas, sean ejecutados por el Sistema Operativo.
- Componentes, que añaden diversas funcionalidades extras al lenguaje.

Se puede apreciar en la Figura 2, cómo el interprete de Gambas, también llamado gbr2, hace de mediador entre la aplicación y el Sistema Operativo, lo cual provoca que Gambas sea un lenguaje de programación portable para la mayoría de las distribuciones de Linux. También funciona sin problemas en la mayoría de los Sistemas Operativos de la familia BSD. Tanto el intérprete como el compilador, están desarrollados bajo el lenguaje C. Se cae de maduro que, para poder ejecutar un programa compilado en Gambas, necesitaremos tener instalado en nuestro ordenador este intérprete.

Básicamente, el Entorno de Desarrollo está compuesto por un resaltador de código fuente, autocompletado automático de sentencias, un diseñador de interfaz gráfica, un debugger integrado y una ayuda offline, entre otras cosas.

Prestaciones

Gambas posee una amplia biblioteca de componentes que nos brinda un gran abanico de prestaciones disponibles a la hora de desarrollar aplicaciones para Linux, lo que lo hace un lenguaje muy apto para el desarrollo de diferentes aplicaciones. Estos componentes, en su mayoría, están programados bajo C/C++, como así también, en el propio lenguaje Gambas, siendo así posible poder desarrollar nuestros propios componentes, si fuese necesario. A continuación, se detalla un listado sobre algunos de los tipos de aplicaciones que se pueden desarrollar bajo Gambas:

- Aplicaciones de texto,
- Aplicaciones gráficas basadas en GTK+/Gnome y QT/KDE,
- Gestión de bases de datos: MySQL, PostgreSQL, SQLite, ODBC,
- Gestión de Procesos,
- Redes: Sockets, HTTP, FTP,
- Trabajo con XML y XSLT,
- Diseño gráfico en 2D y 3D: SDL, OpenGL,
- Empaquetado y despliegue sencillo de aplicaciones en diferentes distribuciones.

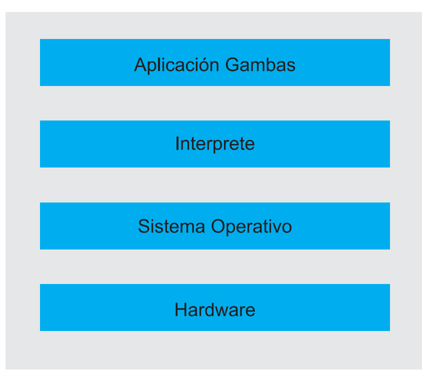


Figura 2. Posición del Interprete de Gambas en un Sistema de Cómputo

Arquitectura

Sin duda alguna, Gambas es un excelente lenguaje de programación por todas sus características y facilidades que le ofrece al programador, pero una de las más importantes es la portabilidad de código, brindada gracias al intérprete. Todo el funcionamiento interno para que un programa escrito en Gambas pueda ser ejecutado en un Sistema Operativo, se debe organizar en lo que se denomina Arquitectura del Lenguaje. Podemos ver un esquema de la Arquitectura de Gambas en la Figura 3.

Pasemos a explicar brevemente su funcionamiento y cómo la arquitectura entra en acción a la hora de hacer un proyecto en Gambas. Principalmente tendremos que encargarnos de codificar nuestro programa bajo Gambas, el cual estará compuesto por nuestros archivos de código fuente como lo pueden ser archivos con extensión `.class` para las clases, archivos `.form` para nuestros formularios, y tal vez algunos archivos adicionales, como pueden ser archivos de imágenes `.tif` o algún otro que sea utilizado en nuestro proyecto. Todos estos archivos, deben estar incluidos en el mismo directorio donde se ha creado el proyecto. Como segundo paso, procedemos a compilar nuestro programa. El compilador utilizado se llama *gbc* y él es el que se encargará de generar, a partir de nuestros archivos de código fuente, un programa que pueda ser ejecutado por el intérprete. El archivo resultante estará codificado en lenguaje Gambas bytecode. Supongamos que somos unos expertos programadores, y nuestro código no contiene ningún error, por lo que obtendremos una compilación exitosa. Acto seguido, el compilador, una vez finalizada su tarea, se ocupará de llamar al archivador, que es el encargado de agrupar todos nuestros archivos recién compilados en un solo archivo, lo que nos proporcionará el archivo ejecutable final. Aquí es donde empieza la tarea del intérprete. Una vez obtenido el archivo ejecutable final, el intérprete se ocupará de analizar y entender cada línea de codigo bytecode generada por el

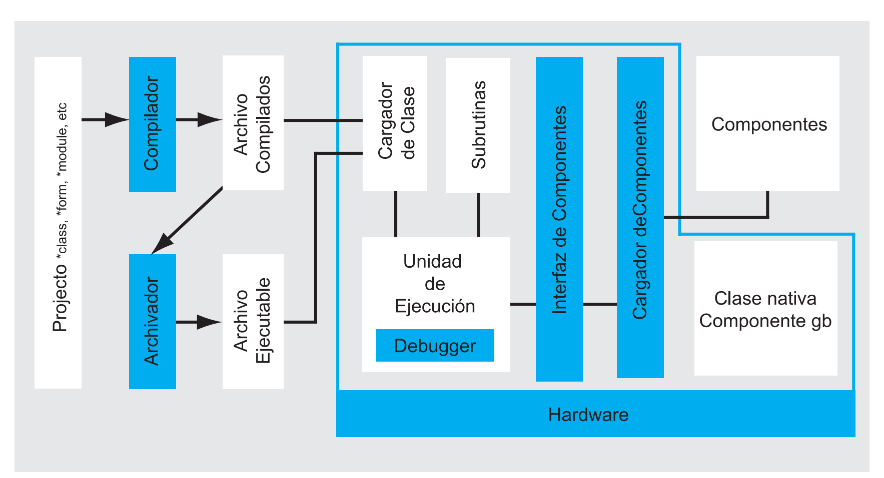


Figura 3. Esquema de la arquitectura del lenguaje gambas

compilador, y así podrá ejecutar nuestro programa en el Sistema Operativo.

En la Figura 4 podemos contemplar un resumen gráfico de las fases de desarrollo de una aplicación en Gambas.

Programación orientada a objetos

Como mencioné anteriormente, Gambas utiliza BASIC como lenguaje base para el desarrollo de aplicaciones, pero añadiéndole exenciones de Programación Orientada a Objetos. BASIC es un lenguaje de paradigma del tipo estructurado, es decir, que el programa empieza en un determinado punto, y sigue bifurcándose dependiendo de las subrutinas que éste tenga, como así también los bucles, las condiciones, etc. Este paradigma utiliza únicamente tres estructuras básicas: secuencia, selección e iteración; siendo innecesario y no permitiéndose el uso de la instrucción o instrucciones de transferencia incondicional, como lo son GoTo, Return, Exit Sub, etc. El paradigma estructurado es sumamente eficiente para proyectos de mediana o pequeña envergadura, ya que si se aplica a proyectos de desarrollo de software más grandes, es muy probable que se produzca lo que en la jerga comúnmente se conoce como programa tallarín. Los programas del tipo tallarín, se destacan por tener infinidad de funciones y subprocedimientos anidados, como así también, muchas líneas de código en su programa principal, lo cual hace que estos programas sean difíciles de mantener. Se le otorga el nombre de programa tallarín, ya que se dice que seguir el hilo de estos programas, es similar a querer seguir un tallarín en un plato de fideos. Las estructuras del programa principal y las llamadas a subrutinas de un programa del tipo tallarín se ilustran en la Figura 5, donde las flechas rojas indican el flujo del programa, y las azules las llamadas a procedimientos y funciones.

Para tratar de corregir el problema de los programas tallarín, a mediado de los 70' se introdujo un nuevo concepto: la POO (Programación Orientada a Objetos). A diferencia del paradigma estructurado, donde el programa se piensa en las *acciones* que éste tiene que realizar, la POO se centra en establecer los objetos *reales* que se necesitan para resolver un problema determinado, y cómo éstos interactuarán entre sí. Aparte de Gambas, se pueden mencionar como ejemplos a C++ y Java como lenguajes con orientación a objetos. Este paradigma trae muchos conceptos nuevos, tales como las clases, la abstracción, la encapsulación, el polimorfismo y la herencia, entre otros. A razón de que estos temas son nuevos para aquel que nunca programó en un paradigma orientado a objetos, nos llevaría casi una entrega entera (o más) de este curso para poderlos cubrir con detalle, por esa razón solamente explicaré las características de los objetos, que son conceptos que nos alcanzan para poder programar en Gambas sin ningún inconveniente. Vale aclarar que Gambas posee la ventaja de darle a elegir al programador el enfoque deseado, pudiéndose desarrollar un programa orientado a objetos o utilizando programación *pseudoestructurada*. Se dice que es un enfoque pseudoestructurado, ya que si el programador aplica un enfoque estructurado a su código, pero desarrolla una aplicación con interfaz gráfica, ésta deberá utilizar los controles gráficos que provee Gambas (botones, formularios, cajas de texto, etc.) y ellos por naturaleza propia, son objetos. Este concepto es un claro ejemplo de la programación multiparadigma, ya que mezcla dos paradigmas totalmente diferentes en una sola aplicación, en este caso, el paradigma estructurado y el paradigma orientado a objetos.

Propiedades, métodos y eventos

Los objetos poseen tres características fundamentales que los conforman: las propiedades, los métodos y los eventos. Para poder explicar estos tres conceptos, veámoslo primero con un ejemplo de la vida real y luego alguno específico del área de la programación. Pongamos como el ejemplo más fácil de comprensión, el *objeto* persona.

Una persona tiene características que la distinguen de cada ser humano, y son propias de cada persona en particular. Estas características pueden ser el color de pelo, la estatura, el color de ojos, su nombre, etc. Todas estas características, son propiedades de la persona, y están definidas por distintos valores.

Ahora bien, cualquier persona que esté totalmente sana, puede realizar acciones como correr, saltar, caminar, estudiar, leer, etc. Estas funciones que la persona puede realizar se denominan métodos.

Supongamos que nuestra persona está leyendo un libro, y de repente aparece una mosca. Esta mosca, se encarga de molestar a nuestra persona y por si fuera poco, la pica. Nuestra persona, ante la picadura, reacciona buscando el veneno para moscas y matando a la mosca en cuestión. Las reacciones que la persona tiene

Figura 4. Proceso del Desarrollo de una aplicación Gambas

ante un determinado suceso (la picadura de la mosca) se denominan eventos.

Ahora pasemos a a un ejemplo orientado a la programación. Queremos realizar un editor de texto, y entre todos los objetos que tenemos, se encuentra un formulario, que contendrá a nuestro editor de texto, con sus botones, barra de herramientas, etc. Este formulario tiene distintas propiedades, como pueden ser su título, su tamaño, su color, su posición, etc. Todas estas propiedades, pueden ser modificadas por el programador, como así también, si el programador lo permite, pueden ser modificadas por el usuario. También, nuestro objeto formulario tiene diferentes métodos, entre los cuales se encuentra el método mostrar, el método eliminar, el método mover, el método refrescar, etc., que se encargarán de realizar una determinada acción, como por ejemplo, mostrar el formulario en la pantalla. Por último, nuestro formulario, por ser un objeto, contiene también eventos que lo caracterizan. Algunos de estos eventos pueden ser el evento clic, que se activa cuando alguien hace un clic sobre el formulario, un evento MouseMove que se activa cuando se mueve el ratón por encima del formulario, etc.

El modelado de los objetos se realiza en lo que se denominan clases. Una clase especifica como debe ser ese objeto, es decir, los métodos que va a tener, sus propiedades, etc. Se dice que una clase es una abstracción de un objeto, ya que me dice cómo va a ser este objeto, pero no es un objeto en sí. Podemos citar el ejemplo de un plano de una casa. El plano (clase) es una representación abstracta de la casa (objeto). El plano me indica las medidas de la casa, el terreno, la cantidad de habitaciones, etc., que son las características del objeto casa. En la POO cuando se crea un objeto, se dice que éste es instanciado, y se ha de especificar de qué clase es el objeto instanciado, para poder definir concretamente sus características.

Pasos a seguir...

Aplicando todos estos conceptos en el desarrollo de un programa con interfaz gráfica en Gambas, los pasos que tendremos que seguir son los que se detallan a continuación:

- Crear un nuevo Proyecto.
- Diseñar y crear la interfaz gráfica.
- Asignarle las propiedades que queramos a los objetos, para obtener el aspecto y comportamiento que nosotros deseamos.
- Codificar los eventos para que el programa realice una determinada acción dependiendo del evento que se haya disparado (por ejemplo, en un editor de texto, en el botón *Guardar* codificaremos el evento clic para que cuando el usuario presione ese botón, se realice la acción de guardar el documento). Estos eventos que codificamos, pueden contener distintos métodos incluidos con las acciones que queramos que realicen los diferentes objetos ante una determinada acción.

¡Manos a la obra!

Dejemos un poco la teoría de lado y pasemos a la práctica. Ya hemos adquirido los conocimientos teóricos fundamentales para poder empezar a desarrollar nuestras aplicaciones bajo Gambas. Sin embargo, a lo largo del curso se irán incluyendo algunos otros conceptos que sean necesarios para la correcta comprensión de los temas dados. Si no te han quedado muy claros algunos conceptos, no te preocupes, con la práctica se te irán sacando esas dudas. Este curso tiene como objetivo lograr que el lector pueda desarrollar aplicaciones GUI bajo la plataforma de Gambas, en conexión con algún Gestor de Base de Datos, como lo puede ser MySQL o SQLite, según sea conveniente. No es requisito excluyente tener conocimientos previos en el lenguaje BASIC o en algún otro lenguaje de programación, ya que va dirigido al publico en general. No se aconseja saltarse ninguna de las entregas, ya que cada una engloba a la anterior, introduciendo nuevos conceptos y adquiriendo un aprendizaje progresivo.

Organización de un proyecto

Comencemos explicando cómo se organizan los archivos en un proyecto. Un proyecto en Gambas se divide en 3 partes fundamentales:

- *Formularios*: aquí es en donde se crea la interfaz gráfica que, en un futuro, interactuará con el usuario, logrando que nuestra aplicaciones sean más amigables. Los archivos utilizados para contener a los formularios tienen extensión `.form`.
- *Módulos o Clases*: aquí se concentrará la inteligencia de nuestro programa, es decir, estará plasmado el código fuente encargado de realizar las tareas que nosotros le indiquemos. Los archivos de este tipo tienen extensión `.class` para las clases y `.module` para los módulos.
- *Archivos Varios*: aquí estarán todos los archivos *externos* que utilizamos en nuestro proyecto. Como ejemplo se puede mencionar, documentos de texto, base de

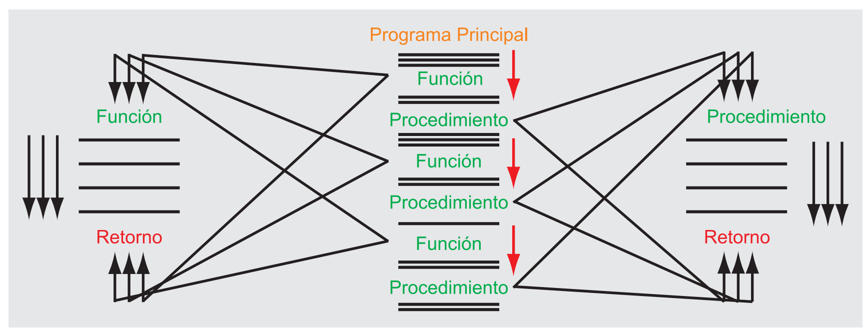


Figura 5. Programa del tipo tallarín

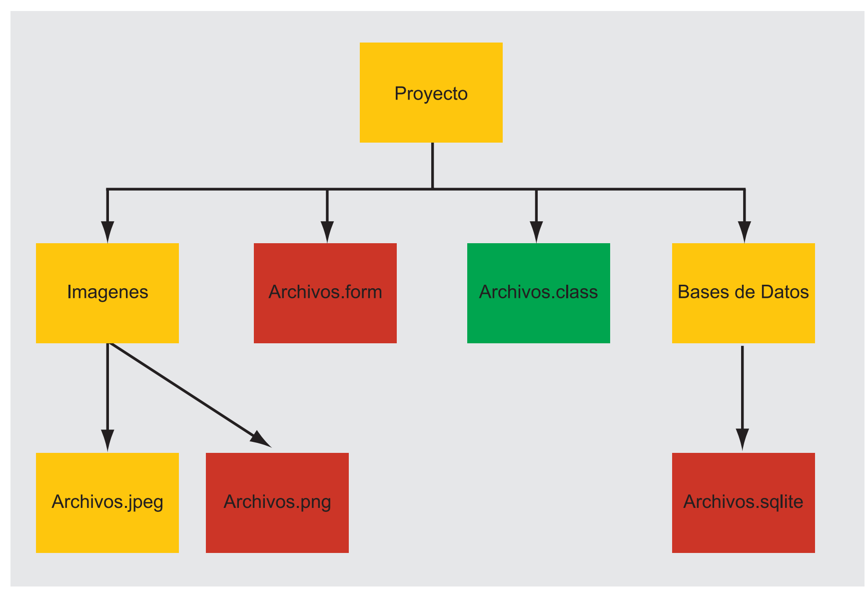


Figura 6. Vista de directorios de la Organización de un Proyecto Gambas



Figura 7. Ubicación de Synaptic en Linux Ubuntu

datos, imágenes, etc. Estos archivos no contienen código fuente Gambas.

Es necesario aclarar que es condición tener todos estos archivos dentro de un mismo directorio raíz, de lo contrario, nuestro programa contendrá errores. Esta situación se puede estudiar en la Figura 6, donde vemos que el directorio raíz contiene a todos los demás archivos del proyecto. Para lograr una mejor organización, se han separado los archivos de imágenes y de base de datos en carpetas que lo distingan más fácilmente.

Figura 8. Pantalla de bienvenida de Gambas

Instalación de Gambas

Sin el IDE de Gambas y su intérprete no podremos desarrollar ningún programa, es por eso que ahora aprenderemos a instalar Gambas en una PC para poder empezar a programar. Debido a la facilidad y popularidad de Ubuntu, es la distribución que pondremos como ejemplo. Como ya había mencionado antes, la mayoría de las distribuciones Linux soportan Gambas sin ningún inconveniente, por lo que si utiliza otra distribución, puede consultar la documentación de cómo instalarla. También en Internet existen diversos blogs que explican de una manera muy sencilla, cómo instalar Gambas. Aprovecharemos la facilidad que nos provee Ubuntu para la instalación de aplicaciones, mediante Synaptic. Primero, nos ubicaremos en nuestro escritorio y abriremos el Gestor de Aplicaciones *Synaptic*, ubicado en *Sistemas ->Administración ->Gestor de Paquetes Synaptic*. Ver Figura 7.

Nos pedirá que ingresemos nuestra contraseña de usuario para poder ejecutar Synaptic. Una vez abierto, nos situaremos en la barra superior y haremos clic sobre el icono que dice *Buscar*. Una vez abierta la ventana de búsqueda, escribimos `Gambas2`, y presionamos la tecla *Enter* para que comience la búsqueda de los paquetes o bien, hacemos clic sobre el botón *Buscar*. La búsqueda de los paquetes puede tardar unos cuantos segundos. Una vez localizado el paquete `Gambas2`, haremos clic derecho sobre el paquete y tildaremos la opción *Marcar para Instalar*. Automáticamente, se nos disparará una ventana informándonos de que el paquete `Gambas2` necesita otros paquetes más que dependen de él para su correcto funcionamiento. Esto es cierto, ya que tal vez no tenemos instaladas todas las librerías que necesita Gambas, por lo que presionamos el botón *Aceptar*. Es importante destacar que al tildar el paquete `Gambas2`, se nos seleccionará también el `gambas-runtimes` que, básicamente, es el intérprete que necesitamos para que nuestros programas puedan funcionar. Luego, una vez que tenemos todos los paquetes necesarios listos para la instalación, solamente nos queda ir a la barra superior de la pantalla y hacer clic sobre el botón *Aplicar*. Mágicamente, Gambas se descargará, se instalará y se autoconfigurará para que podamos empezar a desarrollar nuestra aplicaciones. ¡Excelente!

Iniciando Gambas

Para iniciar el IDE de Gambas, simplemente nos dirigimos al menú *Aplicaciones -> Programación -> Gambas2*. Si por alguna razón Gambas nos ha configurado incorrectamente, cosa que comúnmente no suele pasar, nos dirigimos a la consola de nuestro sistema y tipeamos `gambas2`, y luego presionamos la tecla *Enter*. Así, ya tendremos que estar iniciando Gambas sin ningún inconveniente.

Creando un proyecto

Una vez iniciado Gambas, obtendremos en el escritorio una pantalla como la que se muestra en la Figura 8.

Este asistente nos guiará a través de la creación de nuestro proyecto. Hacemos clic sobre el botón *Nuevo Proyecto...* Como

paso siguiente, el asistente nos dará a elegir el tipo de proyecto que vamos a encarar, si es una aplicación gráfica, una aplicación de consola, una aplicación web, etc. Notemos que tendremos 3 tipos distintos de aplicaciones gráficas. Ésta es una característica interesante que nos provee Gambas y lo pasaré a explicar a continuación. Como sabemos, existen distintos gestores de ventanas para Linux, ya que se encuentran Gnome y KDE entre los más populares, los cuales utilizan para su funcionamiento las librerías GTK+ y QT respectivamente. Estos gestores de ventanas son sumamente diferentes, en lo que respecta a su aspecto visual. Gnome se destaca por tener una apariencia más sobria, pero equilibrándolo con una gran estabilidad y rapidez, por lo cual se recomienda para los equipos más antiguos. Por el contrario, KDE se destaca por poseer un aspecto visual mucho más lindo, pero sacrificando un poco de rendimiento a partir de esta característica, por lo que se sugiere utilizarlo en equipos que posean gran cantidad de memoria disponible. Esto hace que, si nosotros queremos realizar una aplicación para los distintos gestores de ventanas, tengamos que diseñar dos interfaces visuales, una para Gnome y una para KDE. Gambas, nos salva de ese problema. Es por eso que, si solamente queremos que nuestra aplicación tenga el aspecto visual de KDE, elegimos la opción *Aplicación Gráfica QT*, por el contrario, si queremos que nuestra aplicación tenga el aspecto de Gnome, elegimos la opción *Aplicación Gráfica GTK+*, pero aún mejor, si queremos que nuestra aplicación pueda variar entre KDE y Gnome, elegimos la opción *Aplicación Gráfica*, que nos permitirá cambiar el aspecto visual de nuestra aplicación, sin necesidad de agregar, modificar o eliminar ni una sola línea de código. Esta opción es sumamente interesante, y hace notar la gran potencia y flexibilidad que Gambas le brinda al programador. Para nuestro ejemplo, elegiremos la opción *Aplicación Gráfica*, siendo neutral al gestor de ventanas, y haremos clic sobre el botón *Siguiente*. Seguidamente, se nos presentará el dialogo para que indiquemos la ruta de almacenamiento de nuestro proyecto, elegimos la ruta deseada, que puede ser nuestra carpeta personal por ejemplo, y nuevamente hacemos clic sobre el botón *Siguiente*. Ahora, tendremos que ponerle un nombre a nuestro proyecto y un título. El nombre del proyecto no podrá contener espacios ni caracteres especiales. Por último, apretamos sobre el botón *Ok* para terminar la creación del proyecto. Se nos mostrará un mensaje de que el proyecto se ha creado con éxito y luego de confirmar ese mensaje, se nos abrirá automáticamente el IDE de Gambas.

Reconociendo el IDE

El aspecto brindado por el IDE de Gambas es muy similar al de Visual Basic, pudiéndose lograr que a los programadores de Visual Basic les sea más fácil familiarizarse con el entorno. El aspecto del IDE puede verse en la Figura 9.

Para poder trabajar más cómodamente, pasemos a personalizarlo un poco. Nos dirigimos al menú *Vistas*, que se encuentra en la barra de menú, y activamos las opciones *Caja de Herramientas* y *Propiedades*. Se nos aparecerán a la derecha de la pantalla, las respectivas ventanas. Ahora bien, procedamos a identificar las diferente partes del IDE:

- *Barra de menú*: Allí encontraremos accesos directos a las herramientas más utilizadas, como el botón de ejecutar proyecto, abrir y guardar proyecto, etc.
- *Explorador de Proyectos*: Esta vista nos proporcionará todos los archivos que sean utilizados en nuestro proyecto. Se clasifican en Módulo, Clases, Formulario y Datos, siendo este último variedad de archivos como documentos de texto, imágenes, etc.
- V*ista de Edición de Formulario*: Como su nombre lo indica, nos proporciona la vista de cómo está quedando el aspecto de nuestros formularios.
- *Consola*: Tiene como finalidad informarnos a partir de determinados sucesos que surjan en nuestra aplicación, como advertencias, errores, etc. También tiene una pestaña llamada *Identificadores*, donde se podrán ir visualizando en tiempo de ejecución, el valor de nuestras variables.
- *Cuadro de Propiedades*: Aquí establecemos las propiedades de nuestros objetos/controles utilizados.
- *Cuadro de Controles*: Nos provee diversos controles para insertar y utilizar en nuestras aplicaciones, desde botones de comando, hasta controles de Redes. Estos controles se pueden extender, si agregamos los componentes correspondientes.

Nuestra primera aplicación

Nada mejor que un ejemplo funcional para poder aprender y fijar mejor los conocimientos dados en un lenguaje de programación. En este apartado, desarrollaremos nuestra primera aplicación, que se encargará de proveernos información acerca del usuario que la ha ejecutado. Es una aplicación básica, que no requiere de mucho código pero es muy útil como para ir empezando a acostumbrarnos a programar en Gambas. Luego, en el apartado siguiente, explicaremos cómo crear su instalador y archivo ejecutable, para poder distribuirla, y si nosotros queremos, dejarla a dominio público.

¡Empecemos con la aplicación! Primero debemos crear un nuevo proyecto de Gambas, del tipo *Aplicación Gráfica*, al que le otorgaremos el nombre de *UserInfo*. Una vez estando en el IDE, comenzaremos con la fase de la creación de la interfaz de usuario, así luego podemos escribir el código correspondiente. Seguiremos los pasos explicados en el apartado *Pasos a seguir...* . La interfaz creada será similar a la que se muestra en la Figura 10.

Situándonos en el *Cuadro de Controles*, veremos que tiene diversas pestañas, cada una con un nombre de *Categoría* de los controles. Nos dirigimos hacia la categoría *Container*,

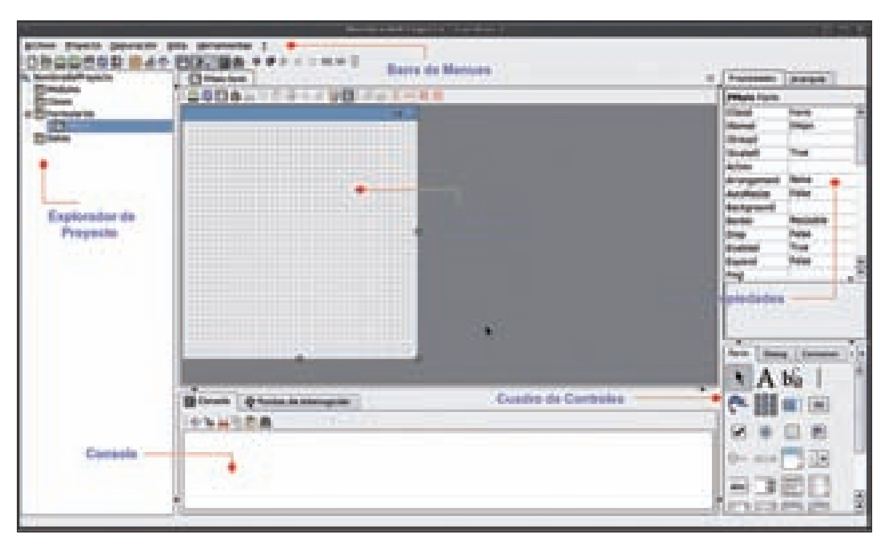


Figura 9. Entorno de Desarrollo Integrado (IDE) de Gambas

que es la encargada de tener los controles que se encarguen de agrupar a otros controles, es decir, no tendrán ninguna funcionalidad propia, solamente una labor estética y organizativa en la interfaz. Una vez ubicado en la categoría *Container*, hacemos doble clic sobre el control *frame* para incluirlo en el formulario, o bien, arrastramos el control hacia el formulario en la posición donde queramos situarlo. Como mencioné anteriormente, solamente este control va a contener a otros controles, no va a realizar ninguna acción en sí. Una vez teniendo el control *frame* en el formulario, lo que hacemos es hacer clic sobre él, para activar sus propiedades. Nos dirigimos hacia el cuadro de propiedades, y buscamos la propiedad *text*, y la cargamos con el valor *Datos del Usuario*.La propiedad *text*, como es de imaginar, contendrá el texto que queremos que muestre el *frame* en cuestión. Podemos ver el cambio en el *frame* en la vista de formulario. Ahora, nos dirigimos hacia la primera categoría, la de los controles del tipo *Form*. Allí, buscamos el control *label* e incluimos uno dentro del control *frame*, que se encargará de contenerlo. Esta acción la realizaremos arrastrando el control *label* hacia adentro del control *frame*. El control *label* o etiqueta, sirve para visualizar algún titulo o cadena de texto que queremos ver en el formulario. Suele utilizarse para poner títulos o información, que es justamente para lo que nosotros lo utilizaremos. Una vez teniendo el control *label* en el formulario, activamos sus propiedades, haciendo clic sobre él. Nos dirigimos hacia el cuadro de propiedades, y buscamos la propiedad *text* y la propiedad *name*. La propiedad *text* es igual a la del control *frame*, sirve para ingresar un texto en el control, mientras que la propiedad *name*, se encargará de asignarle un nombre identificativo al control. Asignamos a la propiedad *text* con la cadena *ID Usuario* y a la propiedad *name* con el valor *lblTituloIDUsuario*. Ahora, procedemos a añadir al *frame* 4 controles *label* más, asignándole las siguientes propiedades:

- Name: lblTituloUsuario Text: "Usuario",
- Name: lblTituloIDGrupo Text: "ID Grupo",
- Name: lblTituloHome Text: "Ruta del Directorio Home",
- Name: lblUsuario Text: (Texto por Defecto),
- Name: lblIDGrupo Text: (Texto por Defecto),
- Name: lblHome Text: (Texto por Defecto),
- Name: lblHora Text: (Texto por Defecto).

Los controles *labels* que tengan la propiedad *text* con el valor *(Texto por Defecto)*, no les asignaremos valores nosotros, sino, en tiempo de ejecución del programa, haremos que estos valores estén cargados con los datos del usuario, por lo que no importa qué valores tengan ahora en el diseño del formulario.

Ahora, añadiremos un control *buttom*, que se encargará de salir de la aplicación cuando hagamos clic sobre él. Lo situaremos en el borde inferior derecho de la pantalla, y le setearemos las propiedades *text* con el valor `Salir`, y la propiedad *name* con el valor `btnSalir`. Nótese que los distintos controles tienen un prefijo establecido, como *lbl* para los *labels*, y *btn* para los *botones*. Esto es para que, a la hora de escribir el código, se nos haga más sencilla la tarea de *reconocer* con qué control estamos trabajando, además de agregarle claridad al código en sí. Todos los distintos controles tienen su propio prefijo.

Como vemos, el diseño de la interfaz gráfica es tan difícil como agarrar un control, arrastrarlo hacia el formulario y situarlo donde lo queramos, y asignarle las propiedades que necesitemos ¡Esto es desarrollo rápido de aplicaciones!

Estaría bien ponerle alguna imagen, para que nuestra interfaz gráfica quede más bonita. Incorporaremos el control *PictureBox*, que se encuentra en la categoría *Form*. Una vez teniendo el control en el formulario, hacemos clic sobre la propiedad *picture*. Se nos abrirá una ventana que sirve para indicarle la ruta de la imagen que queremos añadir, pero nosotros aprovecharemos las imágenes que Gambas trae incorporadas. Hacemos clic sobre la pestaña *Stock*, y seleccionamos la imagen que más nos guste. En el ejemplo, yo seleccioné la imagen *System*. Por último, añadiremos un control especial: el control *Timer*. Este control se encuentra en la Categoría *Special*, y lo que hace es tener una propiedad llamada *delay* que especifica cada cuantos milisegundos tiene que ejecutarse el código que él contiene, es decir, el código que se encuentra dentro del control *timer*. Nosotros lo utilizaremos de la siguiente manera: en Gambas, existe la función *time*, que nos devuelve el horario actual en el formato de *[Horas].[Minutos].[Segundos].[Milisegundos]*. Nosotros nos valeremos de esta función y del control *timer* para que, cada vez que pasa un segundo, el control *timer* se encargue de asignarle la hora actual a un *label* (al label *lblHora*), y así, nos dará la sensación de tener un reloj digital, ya que cada segundo que pase se activará el código del control *timer*, actualizando la hora. Si esto no lo has entendido bien, no te preocupes, cuando lo veas en la práctica te resultará sumamente sencillo. Ponle al control timer las propiedades *delay* con el valor 1000 (es decir, 1 segundo), y a la propiedad *name*, asígnale el valor *tmrTiempo*.

Por fin, hemos terminado la creación de la interfaz gráfica. Ahora vamos a pasar a lo que todo programador le gusta hacer: escribir código. Vamos a empezar codificando el código para que se visualicen en los labels, los datos del usuario. Para ello, tendremos que escribir nuestro código en el evento *Open* del formulario. Este evento es el primero en

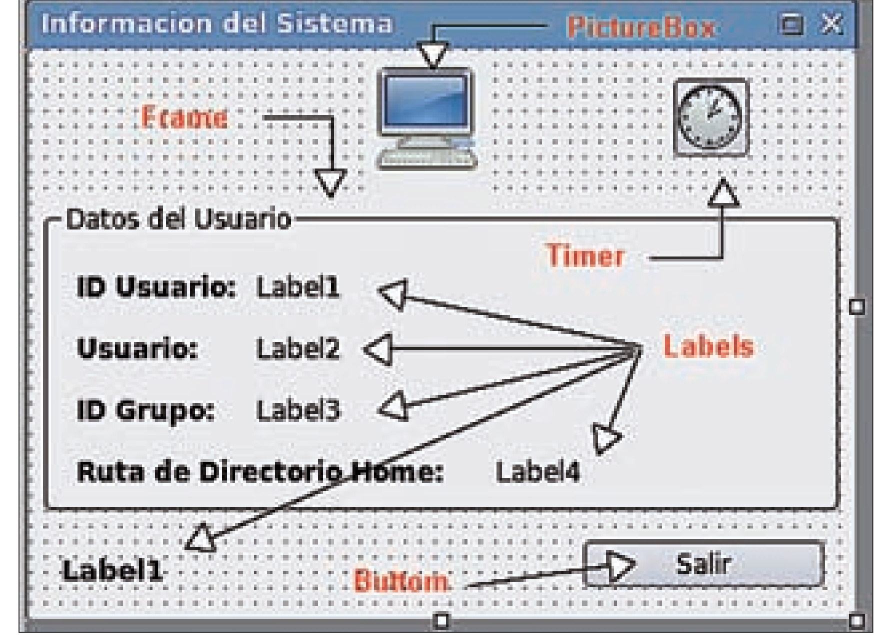
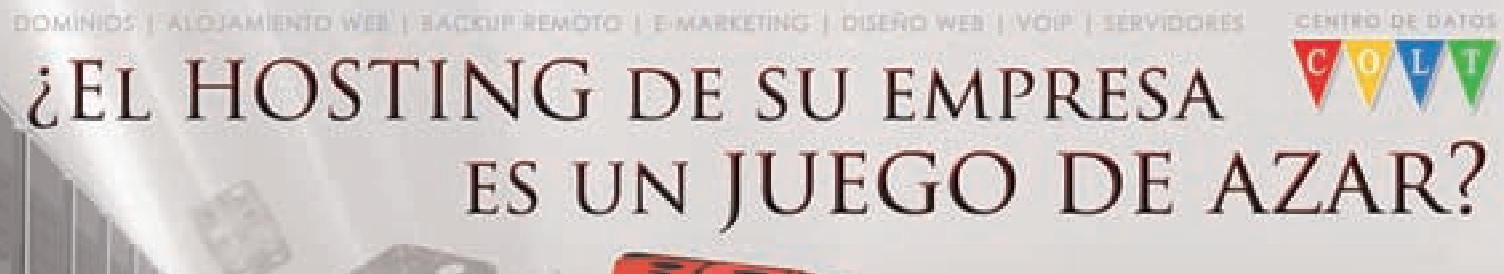


Figura 10. Interfaz Gráfica de nuestro proyecto "UserInfo". Las letras en rojo especifican los nombres de los controles









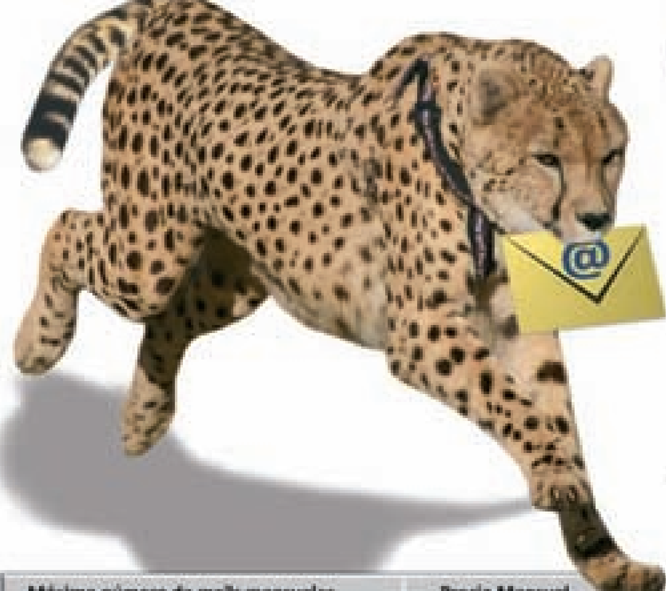



consultorpc.com

SERVIDOR DE ENVÍOS MASIVOS SIN CONSUMO DE SUS RECURSOS ADSL

CPC MAIL RELAY SMTP

RÁPIDO - POTENTE - SENCILLO - FIABLE



Máximo número de mails mensuales	Precio Mensual
5.000 envíos	6 eur
10.000 envíos	10 eur
25.000 envíos	20 eur
50.000 envíos	35 eur
100.000 envíos	60 eur
250.000 envíos	95 eur
500.000 envíos	165 eur
1.000.000 envíos	245 eur
+1.000.000 envíos	consultar

- Servidor SMTP ilimitado y ancho de banda incluidos.
- Envío ultrarápido, más de 100.000 emails por hora.
- IP inversa y cambio de IP gratuito e instantáneo.
- Soporte SPF, DomainKeys Yahoo, Sender ID Microsoft.
- Escalabilidad hasta millones de envíos mensuales.
- Reenvío automático de direcciones de email fallidas.

consultorpc.com

902 99 89 71 | www.consultorpc.com | info@consultorpc.com

DOMINIOS | HOSTING | SERVIDORES | DESARROLLO WEB
BACKUP | VOIP | MARKETING ONLINE | STREAMING | REDES

*Promoción para nuevos contratos. Máximo 12 meses de regalo. No acumulable a otras ofertas.

activarse cuando un formulario es abierto, y nos servirá para cargar los datos del usuario apenas empieza la aplicación. Accedemos al evento *Open* haciendo clic derecho sobre una parte vacía del formulario (es decir, sin estar ocupada por ningún otro control, de lo contrario accederemos al listado de otro evento). Se nos abrirá un menú contextual donde elegimos la opción *Eventos->Open*. Ahora, podremos empezar a escribir el código para que se obtengan los datos del usuario (ver Figura 11 (A)). Podemos ver que el evento empieza con las etiquetas `PUBLIC SUB Form_open` y termina con la etiqueta `END`. Esta forma de delimitar al código del evento de un objeto con `NombreDeObjeto_Evento` es igual para todos los objetos. Los datos del usuario los sacaremos de una clase estática que nos provee Gambas, llamada *User*. A modo de aclaración, las clases estáticas se utilizan para utilizar datos y métodos a las que se puede tener acceso sin necesidad crear una instancia de la clase, es decir, sin tener un objeto que la represente. Esta clase, tiene las propiedades Group, Home, Id, User, que contienen el número de usuario, la ruta del directorio `home`, el id de usuario y el nombre del usuario, respectivamente. Estudiemos el código fuente del evento *Open* del formulario, que se encuentra en la Figura 11 (A).

Como podemos observar, si queremos asignar un valor a una propiedad de un objeto, en este caso, los valores de *User.Group*, *User.Home*, *User.Id* y *User.Name* a los labels correspondientes, se debe usar el signo de asignación "=". El valor que se encuentra a la izquierda de este signo, es el que especifica quien recibirá el valor asignado por el componente que se encuentra a la derecha de este signo. Es este caso, las propiedades *text* de los *label*, están siendo asignadas con el valor que devuelven las propiedades Group, Home, Id y Name de la clase User. También se puede distinguir que, para acceder a una determinada propiedad de un control cuando estamos escribiendo código, se realiza mediante la sintaxis de *Objeto.Propiedad*, como por ejemplo `lblHome.Text`.

Ahora escribiremos el código de salida de nuestro programa. De esta tarea se encargará el evento *Clic* del *btnSalir*, para que, cuando el usuario realice un clic sobre el, se encargue de mostrar un mensaje en la pantalla al usuario y de terminar la aplicación. Ingresemos al evento *clic* del *btnSalir*, al igual como lo hicimos con el evento *Open* del formulario. Una vez allí, escribimos el código que se muestra en la Figura 11(B). Lo que realizan estas dos líneas de código es, en primer lugar, llamar a la clase estática *message*, más específicamente a su método *Info*, que se encarga de recibir la cadena de texto que queramos que salga por pantalla, que tiene que ir encerrada entre comillas dobles. Observemos que cuando tipeamos la palabra `message` y le colocamos el punto, automáticamente se nos presentan cuatro métodos distintos, que son *Info*, *Question*, *Error*, *Delete* y *Warning*. Estos métodos se eligen según el tipo de mensaje que se quiera mostrar, ya que solamente difiere el icono del mensaje que se mostrará en pantalla. Por ejemplo, si se quiere avisar que ha ocurrido un error en la aplicación, usaremos el método *Error* de la clase *message*. En nuestro caso, usamos el método *Info*, ya que solamente le agradeceremos al usuario por utilizar nuestro programa. Queda como tarea para el lector revisar los distintos métodos de la clase *message*, a fin de poder distinguir sus diferencias. Una vez codificado el mensaje, solamente lo que hacemos es llamar a la función `QUIT` que se encargará de terminar nuestro programa.

Como última instancia, codificaremos el evento *timer* del control `tmrHora`. Este evento es el que se disparará cada vez que pase el tiempo especificado en la propiedad *delay*, en nuestro caso, ya lo habíamos asignado con 1000 milisegundos, es decir, 1 segundo. Este código actúa de la siguiente manera: la función *time*, como había explicado anteriormente, se encarga de devolvernos la hora actual del sistema. Nosotros, al ejecutarla cada un segundo en el control *timer*, haremos que simule a un reloj digital, ya que se actualizará el valor *text* del control `lblhora`, cada vez que pase un segundo, y se mostrará por pantalla. Abra el evento *time* y escriba su código como se muestra en la Figura 11 (C).

Ahora llegó el momento tan esperado, probaremos nuestra aplicación. Si no obtuvimos ningún error, ésta se ejecutará sin ningún problema, de lo contrario, Gambas nos avisará y hasta no solucionar el error no podemos ver en acción nuestro programa. De ser así, verifique su código fuente con el que se muestra en la Figura 10, las veces que sea necesario, e inspeccione el valor de las propiedades de sus objetos. Para poder ejecutar nuestra aplicación, nos dirigimos a la barra de menú y hacemos un clic sobre el botón *Ejecutar*, que tiene el dibujo de una flecha, o bien, utilizamos el método abreviado, presionando la tecla *F5* de nuestro teclado. El resultado de nuestra aplicación, será similar al que se ve en la Figura 12, obviamente con los datos de tu usuario en particular.

Verifique su correcto funcionamiento, comparando los datos de su usuario con los arrojados en la aplicación. Fíjese también que, al apretar el botón *Salir*, se nos muestra el mensaje en pantalla que habíamos codificado, y luego se cerrará nuestro programa. Es importante destacar que si salimos de nuestro programa mediante la X que se encuentra en el borde superior derecho del formulario, no se mostrará el mensaje de agradecimiento al usuario ya que, cuando apretamos la *X*, se dispara el evento *Close* del formulario y no el evento *clic* del botón.

```
' Gambas class file

PUBLIC SUB Form_Open()
  lblGrupo.Text = User.Group
  lblHome.Text = User.Home
  lblIDUsuario.Text = User.Id
  lblUsuario.Text = User.Name
END                                          [A]

PUBLIC SUB btnSalir_Click()
  Message.Info("¡Gracias por utilizar el Programa!")
  QUIT
END                                          [B]

PUBLIC SUB tmrTiempo_Timer()
  lblHora.Text = "Hora Actual " & Time
END                                          [C]
```

Figura 11. Codigo fuente de nuestro proyecto "UserInfo". (A) Evento Open del formulario. (B) Evento Clic del botón btnSalir. (C) Evento Timer del control tmrTiempo

Distribuyendo la aplicación

Siempre que creamos una aplicación que pueda ser de utilidad para los demás, está bueno poder distribuirla por la red. Para ello, existen diversas páginas que nos permiten subir nuestra aplicación a Internet, otorgarle una licencia, y hacer un sitio web de nuestro programa, para poder mencionar sus características y colocar un link de descarga. Tal vez, el sitio más popular es Sourceforge (*www.sourceforge.net*) que nos provee todas estas prestaciones (y muchas más) para poder publicar nuestro software. Una vez publicado nuestro proyecto en Sourceforge, tendremos que mandar una descripción de nuestro programa, y luego de una revisión, se decide si se aprueba o no el proyecto. Una vez aprobado, tendremos todos los servicios que nos provee Sourceforge para la distribución de nuestra aplicación.

Cada vez que queremos distribuir una aplicación bajo licencia GPL (General Public Licence, Licencia Pública General), nuestro paquete debe incluir los siguientes elementos:

- Archivo ejecutable,
- Código Fuente,
- Instalador.

No es requisito excluyente que no se tengan estas tres partes de nuestro paquete, pero sí se recomienda que se incluyan (el código fuente y el archivo ejecutable son obligatorias). Gambas nos salva de la tarea de tener que recopilar el código fuente y de tener que crear un instalador, ya que ésto lo puede automatizar. Para poder distribuir la aplicación que creamos, nos dirigimos al menú *Proyecto->Crear->Ejecutable/Archivo Fuente.../Paquete de Instalación...* En las opciones *Ejecutable/Archivo Fuente*, simplemente Gambas nos pedirá que le indiquemos la ruta que queremos almacenar nuestros archivos. Al cambio, cuando creamos el Paquete de Instalación, se nos ejecutará un Asistente, guiándonos en el proceso de creación, donde le tendremos que indicar las características de nuestro programa, el menú donde queramos que aparezca, la distribución en la cual va a ser ejecutado nuestro programa (se puede elegir más de una), y otras características. Como vemos, es muy sencillo crear nuestro paquete para poder publicarlo. Una vez creado, lo podemos publicar mediante Sourceforge u otro sitio que nos brinde un servicio similar.

Vale aclarar que una PC que no tenga Gambas instalado, no va a poder ejecutar programas desarrollados bajo este lenguaje, ya que, como se explicó anteriormente, Gambas necesita un intérprete para que las aplicaciones funcionen. Es por ello que cada vez que se distribuya una aplicación, es necesario informarle al usuario que va a instalar la aplicación, que necesita instalar el intérprete de Gambas. Simplemente el usuario debe ejecutar bajo la consola de Linux el comando `sudo apt-get install gambas2-runtime`, o bien, buscando este paquete por medio de Synaptic.

Referencias externas

Les dejo aquí unos enlaces de interés que les pueden resultar de gran ayuda para poder empezar con Gambas y conocerlo un poco más a fondo:

- Sitio Web Oficial: *http://gambas.sourceforge.net/*,
- Documentación Oficial: *http://www.gambasdoc.org*,
- Documentación no Oficial: *http://gambas.gnulinex.org/*.

Es importante destacar que la documentación oficial se encuentra traducida a más de 15 idiomas distintos, entre los que se encuentra el español. Si bien no está completamente traducida, puede resultar de gran ayuda y ser un perfecto material de consulta. Generalmente, la documentación oficial no está destinada a enseñar el lenguaje en sí, sino que está más enfocada a una guía de referencia.

Consideraciones finales

En esta primera entrega del curso de Gambas, hemos recorrido desde la problemática que se tenía anteriormente para el desarrollo de software visual, pasando por un poco de teoría de POO, explorando algunas características de Gambas y llegando a desarrollar nuestra primera aplicación gráfica. Como nota final se recomienda al lector repasar el código fuente del programa *UserInfo*, intercalándolo con los conceptos teóricos de POO y los temas de Propiedades, Métodos y Eventos, para una mejor comprensión. Cualquier duda, pueden escribirme a *jorge.capurro@linuxmail.org* ¡Nos vemos en la próxima entrega!

Sobre el autor

Jorge Emanuel Capurro es estudiante de la Tec. Superior en Programación, carrera dictada en la Universidad Tecnológica Nacional – Facultad Regional Haedo, provincia de Bs. As., Argentina. Principalmente, su área de investigación se centra en los Sistemas Operativos de tipo UNIX y de la programación bajo esta plataforma. Es el creador del proyecto IDEas (*http://ideasc.sourceforge.net*), que es el primer frontend desarrollado bajo Gambas del compilador gcc (*http://gcc.gnu.org*), que se utiliza con fines didácticos. Actualmente se desempeña como programador para la empresa IBM Argentina en el departamento de Applications Management Services, participando de diversos proyectos de desarrollo de software de carácter internacional.



Figura 12. Ejecución del programa

Tu foro en PHP,
sencillo y seguro

David Puente Castro

Todo comenzó un día como otro cualquiera, una amena conversación de Messenger y mi página web casi terminada: "*Escucha, me gustaría añadir a mi web un pequeño foro donde la gente pueda dejar sus comentarios y otros puedan aportar sus correspondientes respuestas. Pero algo sencillo...*". "*Es una buena idea*". "*Gracias, ¿qué me recomiendas?*". "*Programártelo tu mismo...*"



linux@software.com.pl

Hoy en día es muy fácil optar por algo tan potente como PHPBB (*www.phpbb-es.com/*) o IPB (*Invision Power Board*), para crear casi toda una comunidad dentro de nuestro alojamiento. Pero administrar tales sitios puede convertirse en una odisea, y a veces uno sólo precisa un pequeño espacio para recoger las opiniones e inquietudes de sus visitas diarias.

¡Ahora usted puede hacerlo con sus propias manos! Pero no todo es coser y cantar. Muchos aspectos deben ser tenidos en cuenta, el más importante es la *seguridad* y a estas alturas ya sabemos que la *complejidad* es su mayor enemigo.

Recordemos que un foro es una fuente de entrada de datos para usuarios externos, algunos con buenas y otros con malas intenciones (no lo sabemos). Una incorrecta validación de estos datos puede provocar vulnerabilidades y ataques tan conocidos como lo son *XSS Cross Site Scripting* y el no menos nombrado *SQL Inyection*.

Pero ya llegamos con experiencia, y nosotros vamos a prevenirlo.

Requisitos previos

Daremos por supuestos algunos elementos y conocimientos:

- Dispone de su propia página web y tiene conocimientos sobre programación HTML.
- Tiene la capacidad para editar su sitio y alterar el contenido de sus páginas.
- Posee unos conocimientos mínimos sobre el lenguaje PHP o está dispuesto a adquirirlos.
- Su servidor utiliza el servicio MySQL para alojamiento de base de datos.
- No me hará responsable si el resultado no termina siendo de su agrado.

Dado que fue el origen de esta historia, a lo largo del artículo utilizaremos capturas de pantalla de mi sitio web en: *http://set.diazr.com*

Como sugerencia, aquí puede disponer de un ejemplo de hosting gratuito en el que encontrará todas las facilidades para seguir este artículo: *http://www.zymic.com/*. Sus características básicas son las siguientes:



Figura 1. Logo de PHP MySQL

- 6 GB de espacio en disco,
- 50 GB de transferencia,
- PHP 5, 5 bases de datos mySQL,
- Accesos FTP,
- 99.9% de Uptime,
- Sin publicidad.

Tiene también algunas características interesantes como por ejemplo:

- Zymic Hosting Control Panel,
- PHPMyAdmin,
- Webalizer.

Estructura

Tres elementos principales son los que componen nuestro foro:

- *foro.php*: Página que muestra los posts que se van añadiendo y el formulario principal.
- *post.php*: Página con el código necesario para almacenar los mensajes en una base de datos.
- *reply.php*: Página que muestra el contenido de los mensajes y da acceso a sus respuestas.

Utilizaremos también 2 páginas PHP auxiliares que nos proporcionan datos y variables comunes para las que acabamos de citar, además de las funciones de conexión a la base de datos MySQL.

Aspecto

Para los más impacientes mostraré en la imagen número 3 un posible resultado de la página *foro.php* una vez finalizado el trabajo.

No se preocupe, usted verá las fechas y horas adecuadas de publicación de mensajes cuando tenga todo preparado.

Lo que puede observar a primera vista es un sistema de paginación (en la parte superior), cuya función es mostrar solo 20 mensajes principales por página (o la cantidad que usted elija). De este modo logramos que la web no crezca verticalmente hacia el infinito y su lectura se hace todavía más cómoda.

Seguidamente puede ver los mensajes principales numerados en orden ascendente, y correspondientemente tabuladas hacia el interior, las respuestas relativas a cada uno. Si redundantemente usted respondiera a una respuesta de un mensaje, ésta sería tabulada nuevamente bajo la anterior. No se preocupe, lo irá viendo a lo largo del artículo.

Para que no se confunda posteriormente, lo que usted está viendo en realidad en *foro.php* son los asuntos de los mensajes y las respuestas que sus usuarios han ido introduciendo. Si desea ver los mensajes que hay en su interior, deberá hacer clic encima de los títulos y será la página *reply.php* quien realmente le muestre su contenido y le permita hacer su respuesta. Téngalo muy en cuenta, ya que podría ser fuente de muchas confusiones a la hora de saber qué hace realmente cada una de las páginas.

Veamos ahora un aspecto final del formulario de introducción de mensajes o respuestas (Figura 4). Tenemos para destacar varios aspectos importantes. En primer lugar puede ver como los campos *Nombre* e *E-mail* son rellenados automáticamente dado que en la base de datos su existencia será obligatoria.

Por otro lado, puede ver un campo especial que le indica la necesidad de facilitar un *codigo de seguridad*, que es el que observa en la esquina inferior derecha de la imagen. Este elemento es conocido en el mundo de Internet como un *Captcha*. En resumen podemos decir, que es un desafío que se le propone con el objetivo de diferenciar si usted es un persona real o una máquina automática cuyas intenciones no son precisamente las previstas.

Por último, y antes de empezar con la programación en sí, vamos a mostrar cómo lucirían el contenido de los mensajes y sus correspondientes respuestas según la pagina *reply.php* (Figura 5).

Nada fuera de lo común. El contenido del mensaje leído desde una base de datos, y las respuestas que ha obtenido. Abajo se encontaría el formulario desde el que usted podría responder al mensaje en el cual el campo *Asunto* pasaría a ser el mismo que el del *post* principal precedido de un *Re:*.

Debe ser consciente que el aspecto final dependerá totalmente del estilo que usted utilice en su página web. Lo que siempre se mantendrá es la estructura del foro; al menos esa es nuestra intención.



Figura 2. Página de Zymic



Figura 3. Foro en php, efecto final


Figura 4. Formulario de introducción de mensajes o respuestas

Preparando el entorno

Sin más dilaciones vamos a ver nuestros *scripts* de apoyo correspondientemente comentados.

Ahora voy a explicar las variables presentadas en el Listado 1. No hay mucha complicación: las primeras cuatro variables nos facilitan la conexión a la base de datos MySQL que se aloja en nuestro servidor. Las que restan son variables opcionales donde decidiremos si habilitar un filtrado de palabras perjudiciales, si activaremos la posibilidad de banear direcciones IP que en algún momento hayan intentando realizar algún tipo de ataque, si los posts que se produzcan en el foro deben viajar también hacia el e-mail del administrador, o cuántos mensajes serán mostrados en cada página individual.

En el script presentado en el Listado 2 veremos la función de conexión a la BD (Base de Datos) y un apoyo para imprimir errores que será llamado desde otras funciones cuando un evento malo ocurra.

Para disponer de todos los elementos básicos tan sólo nos queda presentar la estructura de la tabla que debemos introducir en la base de datos MySQL. Hoy en día la mayor parte de alojamientos (*hostings*) gratuitos facilitan los soportes necesarios tanto para almacenamiento de base de datos como para la programacion de sus páginas en PHP. Daré por supuesto nuevamente que usted sabe cómo introducir la tabla que presentaremos a continuación (Listado 3) en el lugar correspondiente. En la mayoría de los casos podrá hacerlo a través de una interfaz web bastante amigable, tal vez el gran conocido como *phpMyAdmin*. Si hace sus pruebas con *www.zymic.com* sabrá inmediatamente a lo que me estoy refiriendo.

Los campos apenas requieren de explicación. En cambio, lo que sí debemos explicar, es cómo nuestro foro identifica en colaboración con la base de datos qué mensajes son principales y cuáles son respuestas a los anteriores. Veámoslo con detenimiento.

Absolutamente todos los posts poseen un campo ID (*identificador*) y otro PID (*identificador del padre*). El identificador será único y exclusivo para cada mensaje particular, no debe repetirse dado que también lo utilizamos como *campo clave* en la base de datos. No ocurre así con el PID, que hará el trabajo de clasificación por nosotros.

Aquellos mensajes que sean nuevos, es decir, que no sean respuesta a ningún otro, sino originales, tendrán siempre un PID igual a 0. En cambio, en aquellos otros que sean respuestas, su campo PID será igual al campo ID del mensaje al que responden. Vamos a verlo esquemáticamente para comprenderlo mejor (Listado 4).

Con lo que acabamos de ver debe comprender perfectamente cómo se lleva a cabo la administración de los mensajes. Fíjese ahora por ejemplo en el mensaje número "1". Como puede ver, su *identificador de padre* es igual a cero, dado que no es una respuesta. Debajo de éste se encuentran dos respuestas independientes a este mensaje cuyo ID es único y particular mientras que su PID los relaciona con el mensaje al que responden, en este caso el identificado con el número "25". Y más importante todavía, observe la respuesta "1.2.1", su campo PID es "27" y no "25" como alguno podría suponer. Exacto, estamos ante una respuesta a una respuesta, y por tanto es hija de la anterior y no del mensaje principal.

Todo lo que acabamos de ver, servirá a nuestro motor de búsqueda para indizar correctamente los mensajes y mostrarlos por pantalla de un modo adecuado.

Figura 5. El contenido de los mensajes y sus correspondientes respuestas según la pagina *reply.php*

Creación de un formulario

Vale, nos estamos moviendo con buen paso, pero antes de administrar ningún mensaje, precisamos al menos de un formulario para que nuestros usuarios puedan escribirlos. Éste será practicamente el mismo tanto en *foro.php* como en *reply.php*. Como hemos dicho hace un rato, la única diferencia será el contenido del asunto, que en el formulario de respuesta contendrá un *Re:* adicional. También existe un campo oculto que será diferente en ambos y al que haremos referencia a continuación. El formulario en código HTML para *foro.php* se muestra como está presentado en el Listado 5.

A pesar de ser un formulario simple con varios campos y un botón, existen 3 aspectos importantes a destacar. El primero de ellos en la siguiente línea:

```
<form method=post action=post.php>
```

Aquí hacemos que una vez pulsado el botón *ENVIAR*, la información sea enviada mediante el método *POST* a la página o script *post.php*, cuyo objetivo es tratar los datos para introducirlos posteriormente en la base de datos. La línea que le sigue es:

```
<input type=hidden
   name=pid value=0>
```

Todo mensaje que se introduzca desde *foro.php* será un mensaje nuevo, nunca una respuesta. Como *post.php* precisa de un parámetro PID que le indique a qué mensaje se está respondiendo, con esto logramos que el formulario siempre envíe un cero haciendo constar que no se está respondiendo a nada. Como dijimos, esta línea difiere del formulario que deberá contener la página *reply.php*, en ésta se verá como sigue:

```
<input type=hidden name=pid
value=<?php echo $row["id"] ?>>
```

Listado 1. Iniciando la programación

```
[vars.php]
<?
     $dbHost = "sql.host.com";              // Servidor MySQL
     $dbName = "basedatos";                 // Base de datos
     $dbUser = "blackngel";                 // Usuario
     $dbPasswd = "contraseña";              // Contraseña
     $badWordFilter = 1;                    // ¿Filtrado de palabras?
     $banIP = 1;                            // ¿Banear IP's?
     $emailAdmin = 1;                       // ¿Enviar posts?
     $adminEmail = "black@set-ezine.org";  // Email administrador
     $maxThread = "20";                     // Posts por página
?>
[fin vars.php]
```

Listado 2. La función de conexión a la Base de Datos y un apoyo para imprimir errores

```
[functions.php]
<?
     require("vars.php");
     function dbConnect() {
          global $dbHost, $dbUser, $dbPasswd, $dbName;
          @mysql_connect($dbHost,$dbUser,$dbPasswd)
                            or error(mysql_error());
          mysql_select_db($dbName);
     }
     function error( $error ) {
          echo "<html><head></head><body>";
          echo " <center><h4><font color=#FF0000>
                Error:$error</font></h4></center>
          <p align=center>
          <a href=javascript:history.back();
                        >Regresar</a></p>";
          echo "</body></html>";
          exit;
     }
?>
[fin functions.php]
```

Lo que hacemos desde *reply.php* es enviar a la página *post.php* el identificador del mensaje que se está mostrando en ese momento, que pasará a ser el nuevo PID (*identificador padre*) de la respuesta enviada (recuerde y revise la explicación de cómo se estructuraban los mensajes).

Y para terminar con el formulario nos vamos a centrar en la primera parte relativa a la seguridad de nuestro foro:

```
<input id="security_code"
name="security_code" type="text"
size=6/>
```

Solicitamos aquí un código de seguridad que será obligatorio antes de realizar ningún post. Es en este preciso momento donde entra en juego el elemento *Captcha Security Image*. Esto es un generador de imágenes aleatorias conteniendo letras y/o dígitos que teóricamente sólo una persona humana puede interpretar. Para que pueda continuar, debe descargar los ficheros adecuados desde la siguiente dirección: *http://www.white-hat-web-design.co.uk/articles/captcha.zip*

Puede visitar la página principal si desea encontrar información directa sobre su aplicación práctica, pero no se preocupe, aquí le explicaremos cómo utilizarlo para nuestro objetivo. Es muy fácil, en primer lugar debe descomprimir todo el contenido del fichero `zip` en el mismo directorio en que situará los scripts *foro.php*, *post.php* y *reply.php*. Luego debe incluir en algún lugar dentro de *foro.php* y *reply.php*, preferiblemente cerca del formulario de introducción de datos, algo como lo siguiente:

```
<p align="center"><img src="CaptchaSe
curityImages.php" /></p>
```

El archivo *CaptchaSecurityImages.php*, se encarga de devolver una imagen con un código aleatorio que será el desafío/respuesta para el usuario que desee hacer un post. Este script puede ser configurado para establecer valores como el tamaño de la imagen, pero esto puede verlo por su cuenta para no complicar demasiado el artículo.

Bien, ahora podría poner el código de cada página que se encarga de hacer todo el trabajo sucio y dar la mayoría de las cosas por hechas, pero ese no sería un método adecuado, así que la mejor forma será estudiar el proceso o camino que sigue un mensaje e ir aplicando el código necesario para cada etapa. Esquemáticamente podríamos decir que un mensaje puede llevar dos rutas distintas, presentadas en la Tabla 1.

Debe entender también, después de lo que acabamos de ver, que *reply.php* es una copia de la página *foro.php* a la que se le han hecho las modificaciones pertinentes para mostrar el contenido de un solo mensaje en vez de la totalidad de ellos y cuyo formulario tiene en cuenta el ID del mensaje mostrado para ser el nuevo PID del mensaje respuesta.

Para ir por partes, imagínese que usted de momento no desea la posibilidad de que los mensajes puedan ser respondidos y tan sólo quiere que los posts realizados en *foro.php* se almacenen en la base de datos y sean mostrados correctamente.

Almacenando mensajes

Ya tenemos un elemento muy importante, el de introducción de mensajes. Ahora vamos a ver cómo hace *post.php* para validar los datos de entrada y grabarlos en la base de datos. Además le reservamos una sorpresa... (Listado 6).

Todo esto es en realidad mucho más sencillo de lo que pueda parecer a primera vista. Lo explicaremos en 4 puntos y se dará cuenta como no le engañamos:

- La función `textFilter()` se encarga de buscar palabras que usted no desea estén contenidas en ninguno de los campos y que previamente deberá tener almacenadas en una tabla llamada *badwords* en su base de datos. Esta función solamente se ejecutará si la variable `$badWordFilter` está definida a 1 en *vars.php*, así que recuerde que

Listado 3. La estructura de la tabla que debemos introducir en la base de datos MySQL

```
CREATE TABLE IF NOT EXISTS 'forum' (
     'id' smallint(5) unsigned NOT NULL auto_increment,
     'pid' smallint(5) unsigned NOT NULL default '0',
     'user' text NOT NULL,
     'mail' varchar(40) NOT NULL,
     'asunto' text NOT NULL,
     'texto' text NOT NULL,
     'fecha' timestamp NOT NULL default '0000-00-00 00:00:00',
     'ip' varchar(15) NOT NULL default '',
     'getmail' char(1) NOT NULL default '',
     PRIMARY KEY ('id')
   ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;
```

Listado 4. Campos de campos ID y PID

```
1. → ID = 25, PID = 0
   1.1 → ID = 26, PID = 25
   1.2 → ID = 27, PID = 25
      1.2.1 → ID = 28, PID = 27
2. → ID = 30, PID = 0
   2.1 → ID = 31, PID = 30
      2.1.1 → ID = 32, PID = 31
```

Listado 5. El formulario en código HTML para *foro.php*

```
<form method=post action=post.php>
   <input type=hidden name=pid value=0>
      <a name=postnew>
         <table border=0 cellpadding=0 cellspacing=2 width=430>
            <tr>
               <td width=80 height=18><b>Asunto:</b></td>
               <td width=430 height=18><input type=text name=topic size = 40 maxlength=50></td>
            </tr>
            <tr>
               <td width=80 height=18><b>Nombre:</b></td>
               <td width=430 height=18><input type=text name=name size=20 maxlength=20 value='Anonimo'></td>
             </tr>
             <tr>
                <td width=80 height=18><b>E-mail:</b></td>
                 <td width=430 height=18><input type=text name=email size=40 maxlength=40 value='anonimo@anonimo.
com'></td>
              </tr>
            <tr>
               <td width=80 valign='top'><b>Mensaje:</b><br><input type=submit value="Enviar"></td>
               <td width=430><textarea style='width:100%;' name=msg rows=10 maxlength=2000></textarea></td>
            </tr>
         <tr>
            <td width=80 height=18></td>
            <td width=430 height=18 align="center"><b>Introduzca el codigo de seguridad: </b><input
id="security_code" name="security_code" type="text" size=6/></td>
        </tr>
   </table>
</form>
```

Listado 6. Validando los datos de entrada y grabándolos en la base de datos

```
<?php
   require("vars.php");
   require("functions.php");
      function textFilter($str) {
         global $badWordFilter;
         if ($badWordFilter) {
            $str = htmlspecialchars( $str );
            $result = mysql_query( "SELECT word FROM 
badwords" ) or
            error( mysql_error() );
            while($row = mysql_fetch_array( $result ))
$str = eregi_replace( $row['word'], "", $str );
            return $str;
         }
      }
      session_start();
      if(($_SESSION['security_code'] == $_
POST['security_code']) && (! empty($_SESSION['security_
code'])) ) { dbConnect();
      $ip = $_SERVER['REMOTE_ADDR'];
      $ip = addslashes(textFilter($ip));
      $eip = explode(".", $ip);
      if( count($eip) != 4) error("Intento de 
hacking!");
      $autor = $_POST["name"];
      $mail = $_POST["email"];
      $tema = $_POST["topic"];
      $mensaje = $_POST["msg"];
      $parent = $_POST["pid"];
      $mensaje = nl2br($mensaje);
      if(!empty($autor) and !empty($mail) and 
!empty($tema)) {
         $autor = trim($autor);
         $mail = trim($mail);
         $tema = trim($tema);
         $mensaje = trim($mensaje);
         $tema = htmlentities($tema);
         $tema = addslashes($tema);
         $autor = htmlentities($autor);
         $autor = addslashes($autor);
         $mail = addslashes($mail);
         $mensaje = addslashes($mensaje);
         $tema = ereg_replace( "<", "<", $tema );
         $autor = ereg_replace( "<", "<", $autor );
         $mail = ereg_replace( "<", "<", $mail );
         $tema = ereg_replace( ">", ">", $tema );
         $autor = ereg_replace( ">", ">", $autor );
         $mail = ereg_replace( ">", ">", $mail );
         $autor = strip_tags($autor);
         $mail = strip_tags($mail);
         $tema = strip_tags($tema);
         $mensaje = strip_tags($mensaje, "<br />");
      } else {
         if(empty($autor)) $autor = "Anonimo";
         if(empty($mail)) $mail = anonimo@anonimo.com";
         if(empty($tema)) $tema = "Tema Nuevo";
      }
      if(empty($mensaje)) error("Es que no quieres 
                           escribir nada?");
      else $mensaje = textFilter($mensaje);
      if(!eregi( "[a-z0-9].", $tema )) error( "Que 
                   diablos pones como tema?");
      else $tema = textFilter($tema);
      if(!eregi( "[a-z0-9].", $autor )) error( "Nombre 
                  invalido, pon algo coherente");
      else $autor = textFilter($autor);
      if(!eregi( "^([._a-z0-9-]+[._a-z0-9-]*)
      @(([a-z0-9-]+\.)*([a-z0-9-]+)(\.[a- z]{2,3})?)$", 
      $mail)) error( "Pon un correo valido, o si no, 
                deja el que esta" );
      else $mail = textFilter($mail);
      $sql = "INSERT INTO forum (pid, user, mail, 
              asunto, texto, fecha, ip) ";
      $sql.= "VALUES ('$parent','$autor','$mail',
      '$tema','$mensaje', NOW(), '$ip')";
      $rs = mysql_query($sql) or die("Error al grabar 
un mensaje: ".mysql_error);
      Header("Location: foro.php");
      unset($_SESSION['security_code']);
   } else {
      Header("Location: foro.php");
   }
?>
```

puede desactivarla de momento si no desea complicarse. Más adelante podrá hacer uso de esta opción para evitar que los mensajes contengan cadenas como `http://`, `<script>`, insultos, groserías, y todo tipo de SPAM.

- Cada vez que usted entra en *foro.php* o actualiza la página, un nuevo código de seguridad es mostrado, pero detrás de todo esto, una variable de sesión llamada `$_SESSION['security_code']` es establecida para posteriormente ser comparada con el código que usted ha introducido en el formulario. Esta comparación es la que se realiza después de la llamada a `session_start()`.
- En caso afirmativo, se comprueba que todos los datos obligatorios han sido cubiertos y se filtran todo tipo de caracteres peligrosos mediante las funciones: `trim()`, `htmlentities()`, `addslashes()`, `ereg_replace()` y `strip_tags()`. Recuerde que nunca está de más ser doblemente paranoico.
- Por último, se ejecuta una consulta SQL para almacenar el mensaje y sus datos acompañantes en la base de datos y se regresa a la página `foro.php` donde debería mostrarse el nuevo post.

Tabla 1. Dos rutas distintas de un mensaje

MENSAJE NUEVO	MENSAJE RESPUESTA
Se introduce el mensaje en *foro.php*.	Se introduce el mensaje en *reply.php*
post.php filtra el mensaje y lo graba en la BD	*post.php* filtra el mensaje y lo graba en la BD
foro.php actualiza la lista de mensajes	*foro.php* actualiza la lista de mensajes

Para poner toda la información a la mano del lector, comentaremos dos puntos adicionales más.

Observación número uno

Se habrá fijado cómo guardamos la dirección IP de la persona que envía el mensaje. Esto es opcional, pero puede servir de mucha ayuda. Imagínese que revisando los mensajes en su base de datos, se percata que ha habido un intento de ataque XSS o SQL inyection no autorizado. Usted podría anotar la dirección IP de su autor, añadirla a una nueva tabla en su base de datos (por ejemplo: `tbanned`) y programar una función de filtrado similar a `textFilter()` que devuelva el control directamente a la página principal, si grabar el post, si la IP de la persona que envía el mensaje coincide con alguna de las almacenadas en la tabla.

Pero si optamos por hacer uso de ésto, debemos ser especialmente cautos. El desconocimiento puede hacerle creer que la dirección IP es un valor que no puede ser modificado, y eso es totalmente incorrecto. Pruebe a instalar en su navegador Firefox algún plugin como *ModifyHeaders* o *TamperData* y observe la facilidad con la que esta dirección puede ser falsificada.

No podemos evitar que la dirección IP sea falsa o esté *spoofeada*, pero sí al menos comprobar que tiene el formato adecuado y evitar la introducción de parámetros adicionales en la consulta SQL de inserción. Puede comprobarlo nuevamente en el código fuente después de la llamada a `dbConnect()`.

Puede añadir a su tabla tanta información como desee, podría añadir por ejemplo el identificador del navegador que utiliza la persona que realiza el post. Puede obtenerlo en la siguiente variable: `$_SERVER['HTTP_USER_AGENT'])`.

Esto resulta muy útil a efectos de estadísticas, pero tenga cuidado nuevamente, debe realizar un filtrado correcto para evitar desgracias.

Observación número dos

Ya para terminar con *post.php*, y como regalito que le habíamos prometido, si usted desea que los mensajes posteados sean reenviados a su cuenta de correo habitual para tareas de administración. Debe añadir el siguiente código hacia el final, justo después de la última llamada a `mysql_query()` (Listado 7).

Visualización de mensajes

Supongamos ahora que ya tiene varios mensajes en su base de datos, incluso respuestas a esos mensajes aunque todavía no hayamos descrito totalmente *reply.php*. Lo siguiente será mostrarlos en *foro.php*, ya que es la página a la que se redirecciona nuevamente una vez realizado un post.

Precisamos dos elementos: una función de búsqueda de mensajes y respuestas, y un sencillo sistema de paginación que nos permita decidir cuántos mensajes mostrar por página. Vayamos por orden y veamos primero ese sistema de paginación que usted deberá colocar en aquella zona de *foro.php* que más le guste. Podría situarlo por ejemplo en la parte superior de los mensajes como mostramos nosotros para mayor comodidad, o debajo de ellos tal como hace Google después de mostrar los resultados de una búsqueda (creo que ya va comprendiendo de qué va el tema). El código pintaría así como está presentado en el Listado 8.

¿Qué podemos comentar? Bien, lo más notorio es que *foro.php* recibe a través del método

Listado 7. Código que introducimos para que los mensajes posteados sean reenviados a su cuenta de correo habitual

```
[-----]
    $tid = mysql_insert_id();
    $time = time();
    $header = "MIME-Version: 1.0\r\n";
    $header .= "Content-type: text/html; charset=iso-8859-1\r\n";
    $header .= "From: $autor<$mail>\r\n";
    $subject = "Post: $tema";
    $body = "<font face=$fontFamily size=2><b>Tema:</b> $tema - <a
href=mailto: $mail>$autor</a> " . strftime( "%c", $time ) . "<br><br>";
    $body .= "$mensaje<br><br>";
    $body .= "___________________________________________________________
<br>";
    $body .= $REMOTE_ADDR;
    $body .= "<br>Responder aqui: <a href=http://set.diazr.com/$replyURL?
id=$tid target=_blank>${replyURL}?id=$tid</a></font>";
   @mail( $adminEmail, $subject, $body, $header );
[-----]
```

Listado 8. Sistema de paginación

```
[-----]
   <?php
       require("vars.php");
       require("functions.php");
       dbConnect();
       global $maxThread;
       $page = $_GET["page"];
       if ( !$page ) $page = 1;
       $rst = mysql_query("SELECT COUNT(*) FROM forum WHERE pid=0;") or
die( mysql_error() );
       $row = mysql_fetch_row( $rst );
       $totalRows = $row[0];
       $pages = intval($totalRows / $maxThread) + 1;
       echo "<p align=left><b>Pagina:</b> ";
       for ($i = 1; $i <= $pages; $i++) {
          if ($i == $page) echo "<b>[".$i."]</b> ";
         else echo "<a href=\"./foro.php?page=".$i."\">[".$i."]</
a> ";
       }
       echo "<a href=#postnew>< Nuevo mensaje ></a>";
       echo "<hr size=1 color=#01a9c0 width=100%>";
       findposts(0, 0, $page);
   ?>
[-----]
```

Listado 9. Una llamada a findposts()

```
[-----]
<?php
   require("vars.php");
   require("functions.php");
   function findposts($PID, $sp = 0, $pagina) {
      global $maxThread;
      dbConnect();
      $n = 0;
      $n1 = ($pagina - 1) * $maxThread;
      $n2 = $maxThread;
      $res = mysql_query("SELECT * FROM forum WHERE pid='$PID'
      ORDER BY id DESC LIMIT $n1,$n2", $con);
     $num = mysql_num_rows($res);
     if (!$num) {
        mysql_free_result($res);
        return FALSE;
     }
     for ($num--; $num >= 0; $num--) {
        $tam = $sp * 8;
        echo "<p>";
        for($i = 0; $i < $tam; $i++) echo " ";
           $row = mysql_fetch_array($res);
           $n += 1;
           $ttime = date("F j, Y, g: i", $row["fecha"]);
           echo $n.". <a href=\"reply.php?id=".$row["id"].
           "\">".$row["asunto"]."</a> [De: ".$row["user"]. "],
           <small>(".$row["fecha"].")</small></p>\n";
           findposts($row["id"], $sp + 1, 1);
     }
     mysql_free_result($res);
     return TRUE;
   }
?>
[-----]
```

Listado 10. Mostrando el contenido de los mensajes

```
[-----]
<?php
   require("functions.php");
   $id = $_GET["id"];
   dbConnect();
   $result = mysql_query("SELECT * FROM forum WHERE id='$id'", $con);
   if(mysql_num_rows($result) == 1) {
      $row = mysql_fetch_assoc($result);
      echo "<table border=0 cellpadding=0 cellspacing=0 width=80%>\n";
      echo "<tr><td><b>DE:</b>     ".$row["user"]. "</td></tr>\n";
      echo "<tr><td><b>ASUNTO:</b> ".$row["asunto"]. "</td></tr></table>\n";
      echo "<p><b>MENSAJE:</b></p>\n";
      echo "<p align=\"center\"><pre>";
      echo $row["texto"];
      echo "</pre></p>\n";
   }
?>
[-----]
```

GET un parámetro llamado `page` que en la URL podría verse como esto:

```
http://www.mipágina.org/
foro.php?page=4
```

Pero esto no siempre es obligatorio, es de suponer que usted tendrá un enlace en cualquier parte de su página, tal vez un menú a la izquierda o en la cabecera, que lleve a sus usuarios hacia el foro. Este enlace no llevará un parámetro `page`, y es por ello que el *script* anterior se encarga de definirlo a "1" si se da el caso, de este modo mostrará los 20 primeros mensajes de la primera página que casualmente serán los últimos introducidos, ya que como verá a continuación, hacemos lo necesario para ordenarlos por fecha y hora de introducción.

Como pudo ver en la primera imagen de este artículo, lo que hace básicamente esta función es imprimir una cadena como la que sigue:

```
Página: [ 1 ] [ 2 ] [ 3 ] [ 4 ] [ 5 ]
... < Nuevo mensaje >
```

En la que cada eslabón es un enlace a *foro.php* con el valor del parámetro `page` que corresponda , excepto el eslabón de la página que actualmente se está mostrando.

La siguiente línea es muy útil:

```
echo "<a href=#postnew><
   Nuevo mensaje ></a>";
```

Si usted vuelve a revisar el código del formulario verá también una etiqueta `#postnew`. Cuando en una página se muestran 20 mensajes con sus consecutivas respuestas, el formulario tiende a desaparecer cada vez más, escondido hacia el inferior de la página. Si su intención es escribir un nuevo mensaje tendría que utilizar la barra de desplazamiento para llegar al formulario y cumplir con su deseo. Pero nosotros pensamos nuevamente en la comodidad del cliente, y con este pequeño trozo de código creamos un enlace que lo lleva directamente a la zona de introducción de datos.

Por último tenemos una llamada a `findposts()`, que será nuestro motor de búsqueda de mensajes y respuestas. Pero veamos primero sus argumentos:

- PID de los procesos a buscar, ahora es 0, pero cambiará para buscar respuestas.
- Parámetro que nos ayudará a realizar el espaciado a la hora de tabular las respuestas.
- Número de páginas a mostrar. Útil para saber el rango de mensajes a mostrar.

Listado 11. Script PHP que llame a `findposts()` para ver las respuestas que el mensaje ha obtenido

```
[-----]
<H5>Respuestas:</H5>
<?php
   if (!findposts($id, 0, 1)) echo "<H6> Vacio </H6>\n";
?>
[-----]
```

Listado 12. El formulario que debe ir situado en *reply.php*

```
[-----]
<form method=post action=post.php>
   <input type=hidden name=pid value=<?php echo $row["id"] ?>>
   <table border=0 cellpadding=0 cellspacing=2 width=430>
      <tr>
         <td width=80 height=18><b>Asunto:</b></td>
         <td width=430 height=18><input type=text
              name=topic size=40 maxlength=50
              value="RE: <?php echo $row["asunto"]; ?>">
      </tr>
      <tr>
         <td width=80 height=18><b>Nombre:</b></td>
         <td width=430 height=18><input type=text name=name size=20
          maxlength=20 value='Anonimo'></td>
      </tr>
      <tr>
         <td width=80 height=18><b>E-mail:</b></td>
         <td width=430 height=18><input type=text name=email size=40
          maxlength=40 value='anonimo@anonimo.com'></td>
      </tr>
      <tr>
         <td width=80 valign='top'><b>Mensaje:</b><br>
         <input type=submit value="Enviar"></td>
         <td width=500><textarea style='height:auto;'
             name=msg rows=10 cols=46 maxlength=2000>
             </textarea></td>
      </tr>
      <tr>
         <td width=80 height=18></td>
         <td width=430 height=18><input type=hidden
             name=getmail value=no></td>
      </tr>
      <tr>
         <td width=80 height=18></td>
         <td width=430 height=18 align="center">
              <b>Introduzca el codigo de seguridad: 
              </b><input id="security_code" name="security_code"
                  type="text" size=6/></td>
      </tr>
   </table>
</form>
[-----]
```

Ahora vayamos directamente con el código en sí. Se trata de una función recursiva. Comienza buscando un menaje con PID 0, y cuando lo encuentra, busca todos los mensajes cuyo PID es igual al ID del mensaje anterior, es decir, sus hijos. Recursivamente sigue buscando si estas respuestas tienen a su vez otras respuestas. Podría verse como una estructura en árbol (Listado 9).

La primera parte calcula a partir de la página obtenida como tercer parámetro cuantos mensajes debe de mostrar y a partir de cuál hacerlo. Pongamos por ejemplo que en nuestro sistema de paginación hiciésemos clic en el número “4”. El resultado sería el siguiente:

```
$n1 = ($4 - 1) * $maxThread = 3 * 20
= 60
$n2 = $maxThread = 20
```

Por lo tanto se mostrarían los mensajes del 60 al 80. ¿Cómo se hace esto? Veamos cómo se comporta la sentencia SQL:

```
SELECT * FROM forum WHERE pid='$PID'
ORDER BY id DESC LIMIT $n1,$n2
```

Se seleccionan todos los campos para un PID indicado y se ordenan según el identificador en sentido descendente. ¿Pero no habías dicho que lo haríamos por fechas? Piense fríamente, el identificador de cada mensaje es un número entero, único e irrepetible. Esto quiere decir que se irá incrementando cada vez que se introduzca un nuevo mensaje. Si los obtenemos en orden descendente tendremos en nuestras manos los mensajes más recientes, todavía más eficiente que buscar por la fecha y la hora.

Después se entra en un bucle principal que a su vez ejecuta un pequeño bucle como este:

```
$tam = $sp * 8;
for($i = 0; $i < $tam; $i++)
   echo " ";
```

Se encarga de controlar en qué rama del árbol de mensajes nos encontramos, e imprimir un múltiplo de 8 espacios, que equivale en la mayoría de los sistemas a una tabulación simple.

Luego se van imprimiendo los asuntos de los mensajes, seguido de su autor, la fecha y la hora de publicación. Lo más importante de este punto, es ver que el asunto es un enlace a la página *reply.php* que recibe como parámetro el identificador de ese mensaje. De esa forma *reply.php* sabrá qué mensaje debe mostrar y cómo buscar también si el mismo tiene otras respuestas.

Visualizar respuestas

Hasta aquí ya tenemos la mayor parte del trabajo realizado. Recapitulemos los puntos que hemos tratado:

- Creación de un formulario para mensajes, con integración de un código de seguridad *Captcha*.



Sobre el autor

David Puente Castro, alias *blackngel*, es un gran aficionado a la programación y la seguridad informática. Original de Ourense (Galicia) y viviendo actualmente en la provincia de Salamanca dedica la mayor parte de su tiempo libre a disfrutar de la experiencia Linux.

Asiduo escritor de artículos básicos sobre temas de seguridad informática en el E-Zine electrónico S.E.T. (Saqueadores Edición Técnica), ampliamente conocido en el Underground Hispano. Actualmente mantiene su segunda página oficial en: *http://set.diazr.com/*, cuyo principal objetivo, radica en ocupar el lugar que en su momento cubrió la web de Hackemate como repositorio de material sobre temas de hacking.

Su primer encuentro con Linux data ya de hace unos 8 años y fue poco más adelante que descubrió el fantástico mundo de la programación y la subcultura hacker. Participa activamente en wargames como: yoire.com, warzone (elhacker.net) y otros.

Aunque trabaja con ordenadores, la mayor parte de sus tareas diarias no tienen una relación directa con esta temática. Auxiliar administrativo, obligado a utilizar el sistema operativo Windows, tiene la costumbre de llevarse su última Distribución Live a la oficina para seguir disfrutando de su SO favorito, Linux, y desarrollar sus pequeños proyectos.

Dedica el artículo a su pareja y familia por la paciencia que demuestran cuando intenta explicarles qué es lo que realmente está haciendo.

Por lo demás, puede encontrarlo online, prácticamente las 24 horas del día, en las siguientes direcciones: *blackngel1@gmail.com* y *black@set-ezine.org*

- Filtrado de seguridad de campos y grabación de mensajes en la Base de Datos con posibilidad de redireccionamiento al e-mail del administrador.
- Visualización de mensajes (*asuntos*) en la página principal del foro, con sistema de paginación incluído.

Ya nos queda poco. Estudiaremos ahora la forma de mostrar el contenido real de los mensajes y cómo visualizar debajo de éste las repuestas particulares a ese mensaje, que como hemos visto hace un momento también serán enlaces a los correspondientes mensajes.



Figura 6. Mostrando una lista de todos sus artículos

En concreto, lo que vamos a ver es el contenido de la página *reply.php*, a la que llevamos haciendo referencia a lo largo de todo el artículo. ¡Ánimo!

El inicio de *reply.php* comienza exactamente con el mismo código PHP que *foro.php* conteniendo la función `findposts()`. Es lógico que sea exactamente la misma; será utilizada para buscar las respuestas al mensaje mostrado. La única diferencia es el campo PID que le pasamos como primer parámetro, que como repetimos un millar de veces, es el identificador del mensaje actual (es bueno que usted tenga claro todos los conceptos, eche un vistazo de nuevo a la imagen número 3 para ver el resultado).

¿Qué función nos propones para mostrar el contenido de los mensajes? Aquí la tiene (Listado 10).

El código anterior debe situarlo en aquel lugar de su página *reply.php* particular, donde desee que se muestren los mensajes. El script realiza una simple consulta a la base de datos solicitando todos los campos de un ID específico. Con los datos obtenidos se imprime código HTML para dar forma al mensaje.

Solamente nos queda añadir otro pequeño script PHP que llame a `findposts()` para ver las respuestas que el mensaje ha obtenido (Listado 11).

Usted podría adaptarlo a sus necesidades y situarlo donde le apetezca mostrar el árbol de respuestas.

A pesar de haberlo explicado anteriormente mostraré, ya para terminar el artículo, el formulario que debe ir situado en *reply.php* para que se dé cuenta de las pequeñas diferencias con el que ha utilizado en *foro.php* (Listado 12).

El PID de la respuesta que se introduce en este formulario es el ID del mensaje que se muestra en esa misma página.

No olvide añadir nuevamente una imagen Captcha, para que sus usuarios puedan validarse como personas humanas:

```
<p align="center"><img src="CaptchaSe
curityImages.php" /></p>
```

Conclusión

Llegado este punto, ya tiene todo lo necesario para montar su propio foro en PHP, como prometimos al principio de este artículo, *Sencillo y Seguro*.

Para no alargar más este articulo, varios aspectos han sido dejados en el tintero. Como administrador del foro, usted podría necesitar alguna vez borrar mensajes molestos o que no le interese que estén publicados en su página. Para eliminarlos tendrá que acceder directamente a la base de datos y borrarlos literalmente a mano.

Pero piense, ahora que ya sabe cómo visualizar los mensajes, tal vez podría crear una página llamada *foro_admin.php*, que solicite una contraseña, y en caso de coincidir con la almacenada, muestre una lista de todos los títulos de los mensajes seguidos de un pequeño enlace llamado *BORRAR*. Este enlace debería apuntar a un script PHP que ejecute una consulta de eliminación de registros según el ID del mensaje elegido.

Más todavía. Si añade a *foro_admin.php* un cuadro de lista y un botón de formulario, puede lograr ordenar los mensajes según el campo que desee, ya sea nombre, e-mail, asunto, dirección IP, etc... Esto hará más fácil encontrar el mensaje a tratar. Vea un ejemplo de como *SET-Ezine* permite mostrar una lista de todos sus artículos.

Nada más espero que el artículo haya sido de su agrado y que lo ponga en práctica cuanto antes. Lo que haga con sus propias manos será mucho más reconfortante y fiable que lo que pueda obtener de fuentes ajenas. Disfrútelo.



En la red

- PHP:
  http://www.php.net
- MySQL:
  http://www.mysql.com
- Captcha Security Image:
  http://www.white-hat-web-design.co.uk/articles/php-captcha.php
- SET, Saqueadores Edición Técnica:
  http://set.diazr.com y *http://www.set-ezine.org*

LINO Operating System

Lino García Morales

LINOOS es el acrónimo de LINO Operating System o LI NO Operating System, según se lea; y este doble, a la vez que antagónico, significado, pone en evidencia el papel de los sistemas operativos centralizados modernos (como MacOS, Windows, Linux, etc.) en la gestión de un mundo periférico distribuido.



linux@software.com.pl

El parecido fonético de LINOOS (Linus) con Linux, acentúa la similitud filosófica de proyecto de software libre; hecho por y para la comunidad.

Un poco de historia

Mucho han cambiado las cosas desde que IBM lanzó el primer ordenador personal, allá por los 80s. Los procesadores son más rápidos, baratos a la vez que potentes, con arquitecturas orientadas a ofrecer cierto paralelismo; los ordenadores incluyen más de un procesador; los periféricos son inteligentes; las *unidades* de almacenamiento sobrepasan el Tera, y los discos duros son sustituidos por memorias de estado sólido; las comunicaciones son rápidas, extendidas y robustas; y así… un largo etc. que bien podría acabar con el asentamiento de diferentes monopolios tecnológicos que no compiten entre sí, sino que se complementan.

Los primeros ordenadores personales tenían un único procesador (incluso un coprocesador), memoria interna y un conjunto de buses o medios de comunicación (con su correspondiente circuitería) para acoplar los periféricos. De modo que las señales que llegaban o salían al ordenador personal eran procesadas por el propio procesador. Por ejemplo, un PC (Apple optó por incluir su propio gestor de vídeo en sus MACs mientras que los PCs (compatibles IBM) permitían utilizar cualquier interfaz compatible con sus buses en detrimento de la fiabilidad) no tenía salida a vídeo. Era necesario insertar una tarjeta especializada que, a su vez, proporcionaba las señales primarias de vídeo (por componentes, o incluso compuesto o señal de televisión).

Esta filosofía centralizada de interconexión proporcionó un soporte de acceso primario, a través de la BIOS (Basic Input Output System) y mediante un sofisticado sistema de interrupciones, y un soporte mucho más complejo, para el manejo de todos los recursos del sistema, vía sistema operativo. Los sistemas operativos proveían concurrencia (la gestión del tiempo de proceso está directamente relacionada con la capacidad de compartir tiempo de ejecución por diferentes programas. Normalmente un programa *corre* hasta que necesita *esperar* por algún dato. La optimización más simple de esta espera pasa por permitir que, mientras tanto, puedan correr otros programas.

Finalmente, si el sistema operativo garantiza la ejecución *compartida* de diversos programas puede parecer que todos *corren* en paralelo o simultáneamente) a través del sistema de interrupciones y, según fueron evolucionando, se impusieron estrategias de cooperación, tiempo compartido, por prioridades o inclusive combinaciones de ellas (la función principal de los sistemas operativos es proveer al usuario (programador) de una máquina virtual fácil de entender y programar que abstraiga toda la complejidad del hardware).

Según se desarrollaban los diferentes periféricos, por los innumerables fabricantes, los sistemas operativos introdujeron el concepto de driver o controlador de dispositivo para manejar las especificidades de cada elemento y sacarle el mayor provecho.

En definitiva el driver es un programa cuya función principal es proporcionar una capa normalizada entre las funciones de acceso del sistema operativo y las específicas del periférico en cuestión según el fabricante.

En la actualidad no existe dispositivo, por pequeño que fuere, cuyas funciones no sean dirigidas por al menos un microcontrolador. El microcontrolador, a diferencia del microprocesador de los PCs, incluye las funciones asociadas a periféricos en el propio chip, en lugar de acceder a ellas a través de los buses de datos, direcciones y control, y, en general, un conjunto de instrucciones reducido capaz de operar en un solo ciclo de reloj.

El microcontrolador incorpora internamente temporizadores, señales de entrada/salida, tanto digital como analógica, comunicación serie, etc. Los dispositivos más complejos, como los relacionados con audio y vídeo, además incorporan uno o más DSPs (Digital Signal Processor) para manipular señales de gran ancho de banda (audio, vídeo, etc.) y complejidad y uno o más microcontroladores para manipular las señales de control o interfaz. Curiosamente, las tres tecnologías evolucionan con cierta convergencia (Intel, por ejemplo, ha dotado a sus procesadores, desde la serie MMX, de instrucciones propias de procesado de señal; Microchip ha desarrollado la familia dsPIC, una especie de microcontrolador con funciones DSP. No obstante su función, velocidad, precio, potencia, sigue ocupando rangos diferentes).

Este hecho también cambió la forma de comunicación entre los diferentes periféricos con la sustitución de las señales digitales (o incluso analógicas) básicas asociadas a cada periférico por la comunicación serie incluida en estos dispositivos. El USB (Universal Serial Bus) es un ejemplo de ello. A diferencia de las múltiples interfaces físicas del pasado, podemos encontrar cualquier tipo de periférico con comunicación USB. Desde el pequeño ratón, hasta sofisticados discos duros de altísima velocidad y capacidad, impresoras, escáneres, entrada/salida de audio/vídeo, etc. Las funciones de los periféricos siguen siendo las mismas, incluso muy parecidas a la que ofrecía la BIOS, sólo la forma de comunicación entre el microprocesador y los periféricos ha cambiado por un estándar que proporciona incluso la posibilidad de arrancar el dispositivo en caliente (sin apagar y encender el ordenador).

Un ordenador personal hoy día consta de uno o varios procesadores, sin prácticamente controladores periféricos (DMA, Direct Memory Acces; interrupciones, temporizadores, etc.), memoria RAM (Random Access Memory) de alta capacidad y velocidad, memoria Flash (la Flash EPROM es una memoria no volátil grabable in situ) actualizable (que hace las funciones de la BIOS aunque se le suele denominar firmware o, inclusive, alberga al sistema operativo), controlador de vídeo incorporado o un bus de muy alta velocidad para ello (como el AGP, Accelerated Graphics Port) y algún conjunto de buses series normalizados (como el USB, Ethernet, Firewire ó inalámbricos como IrDA, Bluetooth y WiFi). Sin embargo, a pesar de la enorme descentralización, los sistemas operativos mantienen una centralización encubierta.



Figura 1. Página oficial de XML, *http://www.w3.org/XML/*

Arquitectura

El desarrollo arquitectónico de un PC se podría dividir en dos etapas, con una línea divisoria muy borrosa, que, para simplificar aún más las cosas, podríamos llamar: el antes y el después.

La arquitectura correspondiente al antes posee una unidad de proceso que debe intercambiar información con todos los periféricos; cada uno según su protocolo de acceso mientras que la unidad de proceso del después puede estar formada por varios procesadores que se comunican con todos los periféricos con sólo unos pocos protocolos estándares. Ambos son esquemas centralizados; sin embargo, la introducción de cierta inteligencia en los periféricos permite reorganizar este modelo para conseguir distribución y paralelismo.

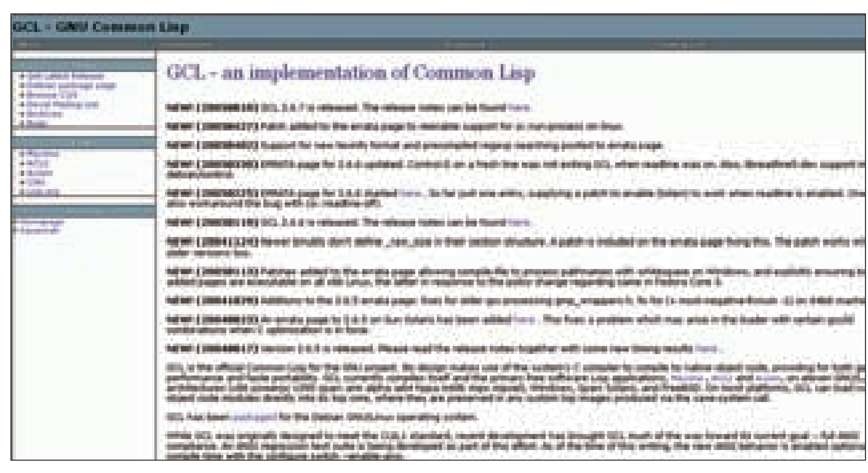


Figura 2. Página del proyecto LISP, *http://www.gnu.org/software/gcl/gcl.html*

Para ello podemos esquematizar el después como una red local, formada por la interconexión de diferentes tipos de redes (según los diferentes protocolos de comunicación), donde cada elemento de la red (punto) tiene determinada función. Para hacer más fácil el análisis consideremos este esquema según el modelo de interconexión abierto (OSI, Open System Interconection) de la ISO (International Standard Organization). Esto nos posibilitará abstraernos de los detalles de las conexiones y concentrarnos en sus interfaces a nivel de aplicación (a este nivel de abstracción se puede considerar que todos los integrantes de la red pueden intercambiar información sin profundizar en cómo lo hacen. El modelo OSI de la ISO descompone la comunicación entre puntos en siete capas compatibles entre sí a nivel de interfaz).

Ahora veamos a todos los puntos como cajas negras (el concepto de caja negra permite ver cada componente del sistema a través de sus interfaces. Sabemos *qué* hace y el protocolo de intercambio de información (qué parámetros y funciones son necesarios y qué resultados esperamos) pero no sabemos *cómo* lo hace) capaces de realizar determinado conjunto de funciones. Cada función o procedimiento consume información, a través de los parámetros o datos de entrada, la procesa, y devuelve unos resultados, a través de los parámetros o datos de salida. En este modelo, cada punto debe publicar sus interfaces (funciones y parámetros de entrada y salida de cada una de ellas) en un formato común. La arquitectura SOA (Service Oriented Arquitecture) es un ejemplo de ello. Los datos de intercambio son formateados en cadenas XML (eXtensible Markup Language). Si cada punto es capaz de publicar sus funciones e interfaces (estas funciones e interfaces, en concepto, son similares a las definidas en la BIOS; sin embargo, es importante un diseño armonioso que pueda evolucionar sin límites. Para ello podríamos tomar el ejemplo del protocolo MIDI (Musical Instrument Digital Interface) que fue capaz (en 1982) de definir un conjunto de mensajes generales para todos los fabricantes y un *mensaje exclusivo* para el uso privado de cada fabricante. De esta manera prepararon al protocolo para aguantar el paso del tiempo; cualquier necesidad posterior es susceptible de insertar mediante un mensaje exclusivo) a toda la red local; cada punto de la red local es capaz de invocar estas funciones. Por ejemplo, la pantalla o display, que tendría incluida la tarjeta gráfica (compatible OpenGL), exportaría una serie de funciones relacionadas con su uso (compatible XWindows). El propio disco duro podría hacerse cargo del sistema de archivos; de manera que pueda publicar sus directorios, archivos, contenidos de archivos e inclusive realizar funciones de defragmentación automática en sus ratos de ocio. De esta manera un editor de texto, por ejemplo, funcionaría independientemente del tipo de monitor, disco duro, teclado, ratón o impresora como una auténtica aplicación distribuido en la que cada punto juega su rol independientemente de cualquier sistema operativo. Por último suponga que denominamos a esta arquitectura LINOOS.

Sistema operativo

Con LINOOS, en esta arquitectura distribuida podríamos cuestionarnos la necesidad de un sistema operativo centralizado moderno. Observe que LINOOS no necesitaría de ningún driver para operar, sólo de las interfaces de cada punto de la red. Analicemos entonces el punto procesador. No todas las aplicaciones requieren del uso de un sistema operativo. Normalmente las aplicaciones diseñadas para microcontroladores no lo hacen y esto es debido a lo limitado de sus recursos y la especificidad de sus funciones. Son sistemas a medida y óptimos y es que el sistema operativo introduce cierta sobrecarga, consume determinada memoria, espacio en disco, etc. ¿A cambio de qué? De garantizar la concurrencia entre diversos programas, la comunicación entre ellos, arbitrar el uso de recursos compartidos, etc.

Las aplicaciones DSP son otro caso donde habitualmente no se utiliza un sistema operativo. Son aplicaciones donde lo más importante es la velocidad y optimización de los recursos y, normalmente, son muy específicas. Lo cual limita el uso de los sistemas operativos a aplicaciones generales donde lo más importante no es precisamente la optimización (hemos sido testigo de cómo los sistemas operativos, que hemos llamado centralizados modernos, devoran espacio en disco, memoria y recursos insaciablemente para proveer un entorno más amistoso de dudosa justificación. Mucho más nocivo en cuanto no dejen opción a usos más austeros).

Sin embargo, es posible aplicar la filosofía anterior para conseguir un procesador LINOOS. En este caso cada procesador puede tener su mini (el término mini es justificado por la reducción de sobrecarga que introduce la gestión de los drivers, y la exclusión de todo lo que no sea estrictamente relacionado con la gestión de concurrencia, comunicación entre procesos y compartición de los escasos recursos de los que dispone) sistema operativo (o gestor de tareas) en su propia memoria flash (observe que no es necesario un disco duro y que el firmware (mini sistema operativo) es auto actualizable a través de su propia interfaz) y exportar, a través de sus interfaces de comunicación, sus servicios.

Observe que un sistema LINOOS sería susceptible de integrar múltiples puntos (en principio ilimitado) con diferente, similar o igual funcionalidad; lo que permitiría crear verdaderos sistemas multiprocesamiento y compartir realmente todos los periféricos (podríamos tener múltiples pantallas, cada una visualizando diferente información; múltiples sistemas de almacenamiento; múltiples impresoras; etc.).

Un paso más allá

El siguiente paso sería la interconexión de diversos sistemas LINOOS; lo cual es prácticamente extensible si existe al menos un punto de la red que pueda servir de pasarela o gateway. Para compartir recursos entre diferentes sistemas LINOOS es necesario proveer a cada componente del sistema de una dirección única.

El uso de direcciones IP (Internet Address) garantiza un enumerador único; pero es posible algún tipo de direccionamiento más simple utilizando técnicas de numeración local para diferenciar cada dispositivo con una dirección única. Lo más simple es emplear algún tipo de código identificador de cada tipo de dispositivo, según su función (agruparlos en familias (algo resuelto muy bien por GM, General MIDI)). Por ejemplo:

- 1 para monoprocesadores,
- 2 para multiples procesadores,
- 3 para microcontroladores,
- 4 para DSPs, etc.

seguido de un identificador de dirección. O algún mecanismo, como el que emplea el I2C, que asigne rangos de direcciones según las diferentes funciones. Así 1.001 correspondería al monoprocesador 1 dentro de la red. Para diferenciarlo de aquellos que pertenezcan a otras redes locales se podría complementar la dirección con la IP del gateway, por ejemplo. En el caso de que coexistiesen múltiples gateways sería necesario diferenciar los puntos según el gateway al que estén conectados.

Lo más importante

Probablemente, con muy poco esfuerzo sería posible crear sistemas LINOOS, a partir de los sistemas operativos habituales; de



Sobre el autor

Graduado de Ingeniería en Control Automático, Máster en Sistemas y Redes de Comunicaciones y Doctor por la Universidad Politécnica de Madrid. Ha sido profesor en Instituto Superior de Arte, Universidad Pontifici.a *Comillas* y la Universidad Meléndez Pelayo.

Actualmente Profesor de la Escuela Superior Politécnica de la Universidad Europea de Madrid y Director del Máster Oficial en Acústica Arquitectónica y Medioambiental. Lidera grupo de investigación transdisciplinar en la intersección Arte, Ciencia, Tecnología y Sociedad. Becas por la Agencia Española de Cooperación Internacional, FUNDESCO, Consejo Superior de Investigaciones Científicas y Universidad Politécnica de Madrid.

Además ha formado parte de las bandas Cartón Tabla y Música d' Repuesto. Discografía: Las palabras vuelven, Se fue, Bags, Flags, Faqs, Fotos d' parque, These little things that keep inside, Mr. Fro (colaboración con Alejandro Frómeta), El eje del mal, Av abuc, Variaciones en la cuerda VOL. I.

Y ha publicado una novela: *Islas*.

lino@madrid.com

hecho, sería lo recomendable. Si es así, entonces ¿qué tiene de gracia este cambio? Lo más importante probablemente de este enfoque es que consigue una total independencia entre aplicación y tecnología. Sería posible sustituir cualquier punto por otro de última generación con tecnologías más rápidas, potentes y estables mientras todo seguiría funcionando. ¿Por qué? Porque sólo cambia el cómo no el qué hace y, mientras no cambien las interfaces, todo seguirá encajando perfectamente (este es el mayor legado del modelo OSI probablemente). LINOOS, visto de otra manera, conseguiría crear sistemas robustos a la obsolescencia tecnológica.

Aunque el XML está tan extendido, sería posible definir los mensajes de intercambio en un metalenguaje como LISP (de hecho existe una relación muy directa entre ambos). De esta manera no sólo se podría transferir información sino también código fácilmente ejecutable en cualquier punto del sistema LINOOS.

Otro aspecto reseñable sería su escalabilidad. Para aumentar la potencia de proceso, por ejemplo, bastaría con aumentar los puntos de proceso que, dicho sea de paso, serían más baratos y simples (la conservación de la escalabilidad sería posible también por la base OSI subyacente en la arquitectura). Sería posible cambiar, inclusive, el tipo de enlace de comunicación, e incluir nuevas tecnologías que pudieran surgir en el futuro, sin que se alterase el funcionamiento del sistema. ¿Por qué? Porque LINOOS trabaja en la capa de aplicación; lo que hace posible sustituir cualquiera de las capas inferiores, varias inclusive, sin influencia en el conjunto.

Otro aspecto interesante, desde el punto de vista de desarrollo, es considerar los mecanismos de comunicación interprocesos independientemente de si coexisten o no en el mismo punto. De esta forma tendríamos un sistema transparente a la arquitectura y los mecanismos de arbitraje de recursos, normalmente controlados por el sistema operativo, estarían distribuidos.

En término de conectores (en este contexto un sistema operativo (kernel), internet o un medio es un conector), considerado como un mecanismo de comunicación que hace posible acoplamientos, LINOOS garantizaría diseños/arquitecturas de alto nivel, reutilización, distribución/paralelización, etc.

¿Sería posible conseguir sistemas más seguros en LINOOS?, probablemente. Con el concepto de caja negra en mente, en LINOOS cada punto de la red es responsable de todas sus funciones, incluida las relacionadas con la seguridad. No es posible formatear un disco duro malintencionadamente, sólo el propio dispositivo puede hacerlo si recibe la orden y coteja que, quien la emite, tiene permiso para ello; mecanismo que se puede activar en un proceso de configuración inicial.

LINOOS converge hacia lo que se va conociendo como tecnología Web 3.0. En este entorno cada dispositivo, por diferente que sea, puede exportar su funcionalidad a la comunidad.

Conclusión

Sólo una última cosa. El éxito de LINOOS se basa en el empleo de estándares libres y abiertos a toda la comunidad y, como puede intuir, LINOOS aún no existe. Lo he empleado como una excusa para discursar acerca del papel de los sistemas operativos actuales y para esbozar una idea, que no por simple, puede ser muy potente. LINOOS no es caos, sino un mecanismo de cooperación que permitiría el desarrollo, pero más importante, el mantenimiento, de aplicaciones complejas. ¡Estáis todos invitados!

Si quieres formar parte de nuestro equipo y crear la revista LiNUX+ DVD con nosotros como:

Autor
Nos gustaría que LiNUX+ DVD fuera una revista realizada por y para los profesionales de GNU/Linux. Por ello, estamos buscando personas con un elevado conocimiento en la materia, expertos en sistema GNU/Linux y a los que les encante escribir. El autor será siempre quien elija el tema.

Corrector
Si la GNU/Linux es tu pasión, conoces en profundidad la gramática y ortografía espa ola y lees el Diccionario de la Real Academia todas las noches antes de dormir, posees un perfil ideal para ser nuestro corrector y corregir los textos antes de que sean publicados.

Betatester
Los betatesters son los que leen los artículos y después opinan sobre ellos antes de que salga la revista. Gracias a esto, sabemos cuáles son los temas más interesantes para nuestros lectores. Si eres uno de nuestros betatesters tu nombre será publicado en la revista. Cuánto más nos ayudes, más puedes esperar de nosotros. ¡Nuestros betatesters son muy importantes para nosotros!

Recuerda: Todo depende de tu voluntad, ¡ nos ayudas cuando tienes tiempo y ganas!

no lo dudes ni un instante, escribe ahora mismo a: es@lpmagazine.org

Urban Terror

Como cada mes, os presentamos los juegos para jugar con vuestros sistemas GNU/Linux, y en este mes comenzamos con uno de los grandes, que aún no habíamos comentado. Digo grande porque sin lugar a dudas es un juego cuidado, con mucho trabajo tras él y que tiene una gran comunidad de desarrolladores y de jugadores. Mes tras mes, parece que los juegos libres de cierta calidad algún día se van a acabar, pero como podéis comprobar cada vez que leéis esta sección, cada vez hay un mayor número de juegos libres y posiblemente la calidad de estos os sorprenda.

Urban Terror pertenece al género de los juegos de disparo en primera persona. De hecho, está basado en el motor de Quake 3. Sus gráficos, son de una excelente calidad y con sólo ver unas cuantas tomas de pantalla apreciaréis la calidad del mismo. El juego es absolutamente gratis, pero su código no está liberado completamente. El motor del juego, el de Quake III, sí es libre, pero todas las modificaciones realizadas sobre él, son propiedad de Frozen Sand, antes conocida como Silicon Ice Development, la empresa desarrolladora. Aunque no sea libre al 100%, sí es gratis, además de multiplataforma, por lo que probablemente interese hasta a los más puristas.



Figura 1. Urban Terror

Aparte comentaros que en realidad, más que de un juego podemos hablar de un mod, es decir, una modificación del juego que no tiene suficiente entidad para decir que es un juego distinto. De hecho, fue nominado en el año 2007 a ser el Mod del año.

A diferencia del juego del que procede, Quake III, Urban Terror es un juego más realista y está orientado a un tipo de jugador que le guste más la estrategia que la acción constante. Podríamos decir que es un juego que te resultará adictivo durante más tiempo. Sus gráficos y escenarios son dignos también de mención, por la cantidad de detalles y la calidad de los mismos.

Los modos de juego que nos ofrece Urban Terror son (todos en inglés, porque creo que los entendéis casi todos y al traducirlos perdería parte de su sentido): Deathmatch (gaming), Free for All (FFA, DM), Team Deathmatch (TDM), Team Survivor (TS), Capture the Flag (CTF), Capture and Hold (CAH), Follow the Leader (FTL), Bomb Mode (BM).

En definitiva, uno de los mejores juegos disponible a día de hoy para GNU/Linux. También está disponible para Mac OS X y para Microsoft Windows por lo que las partidas multijugador las podréis jugar sin problemas contra rivales de estos otros sistemas. Especialmente recomendable para los aficionados a juegos como el mítico Counter Strike o cualquier otro juego similar, más estratégico y menos rápido que Quake III.

http://www.urbanterror.net/



America's Army

A diferencia de muchos otros juegos que comentamos en esta sección, el que nos ocupa en este punto, America's Army, no tiene licencia libre, pero debido a otros muchos factores merece estar en estas líneas.

Hasta hace muy poco, no conocía este juego y cuando comencé a leer sobre él, me sorprendió realmente todo lo que lo rodea. Lo primero que ví, es que era el juego oficial del ejército americano. Este punto me resultó verdaderamente extraño a la vez que llamativo y tras leer un poco más descubrí que en realidad no es sólo un juego, sino un simulador a la vez y que quien lo desarrolla, es el propio ejército de los Estados Unidos.

El por qué se creó un videojuego de este tipo con recursos públicos hay que buscarlo en la falta de reclutas a finales de los años noventa y principios de esta década. Tras intentar de manera negativa varias alternativas, el Teniente Coronel Casey Wardynski consiguió convencer y sacar adelante el proyecto de un simulador de combate en el que el trabajo del equipo, la iniciativa, el liderazgo y sobre todo, ser lo más parecido posible a la realidad, fueran sus ejes.



Figura 2. America's Army

El trabajo fue comenzado por programadores de la Marina estadounidense y su desarrollo gráfico se basó en uno de los mejores motores gráficos existentes, Unreal Engine. Por cierto, quien no conozca este motor sólo decirles que es el motor que también usan otros muchos juegos comerciales como Unreal Tournament, Tom Clancy's – Splinter Cell, Medal of Honor.

El tiempo ha pasado y se han sucedido distintas versiones de este simulador, con versión instalable para GNU/Linux, y alrededor de él se ha creado una comunidad de jugadores y colaboradores que, han conseguido el propósito primero del proyecto, conseguir aumentar el número de reclutas.

Además, el juego es gratuito para los usuarios, por lo que los costes de desarrollo corren a cargo de los fondos públicos del ejército americano.

En definitiva, un juego extraño donde los haya pero de una calidad realmente alta. No es libre pero es gratuito, por lo que no creo que os cueste mucho dedicar un pequeño rato a instalarlo y comprobar la calidad y lo realista que es. Existen versiones para otras plataformas por lo que también podrán probarlo vuestros amigos y conocidos que usen otros sistemas.

http://www.americasarmy.com





Hedgewars

Hace ya muchos números hablamos de dos clones de la saga Worms: OpenLieroX y Wormux. Pasado el tiempo volvemos a traer otro juego inspirado en esta misma saga, Hedgewars. Sin lugar a dudas, uno de los juegos más populares de la década de los noventa, fue esta saga. Su simplicidad, unido a su adictividad, hicieron que la saga se convirtiera en todo un fenómeno de masas. Si os gusta la arqueología informática recomiendo que os paséis por el artículo de la Wikipedia de la saga Worms.

El modo de juego es bastante simple. Una serie de individuos juegan a tirarse todo tipo de municiones para matarse en un escenario cerrado. Si leéis sólo esta frase posiblemente estaréis preguntandoos cómo es posible que algo tan simple sea tan adictivio. Para comprobar esto podéis instalar éste o cualquier de los otros dos juegos libres.

Vamos a comenzar a hablar qué tiene distinto Hedgewars, de los otros dos juegos que os he comentado. La primera diferencia que

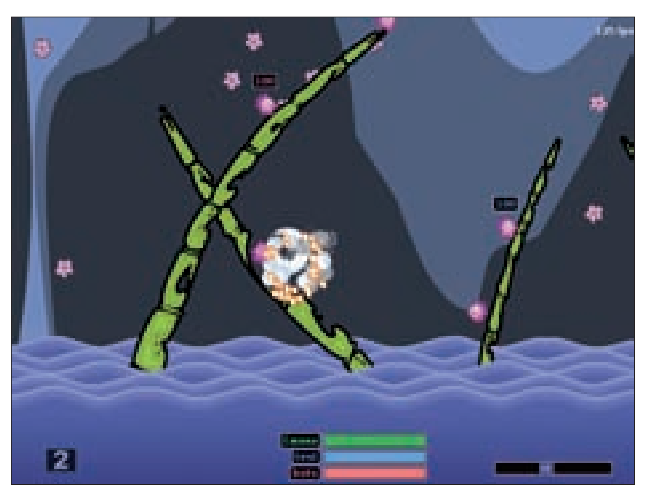


Figura 3. Hedgewars

observamos es que los personajes en lugar de ser worms son erizos. Si lo comparamos con su gran rival, podemos decir que Wormux le gana en variedad de armas y escenarios, pero Hedgewars mejora otros aspectos. Por ejemplo, tiene una inteligencia artificial más compleja, el editor de equipos está más cuidado, es más estable cuando jugamos en red, una física más cuidada y una sensación de mayor calidad cuando estamos jugando. Cuando me refiero a más calidad, me refiero a que destila mejor calidad gráfica, animaciones más suaves y todos esos pequeños detalles que no sabría cómo describirlos pero que se palpan cuando estamos jugando.

Los usuarios de FreeBSD y los de Microsoft Windows también pueden disfrutar del juego. Respecto a los usuarios de GNU/Linux, tenemos paquetes preparados para Debian, Ubuntu y todas las distribuciones que usan el gestor de paquetes APT, para Gentoo y para ArchLinux.

En definitiva, otra alternativa más a un juego ya existen, pero como siempre le explico a la gente que desconoce este mundo del software libre, podemos elegir entre distintas alternativas, cada uno adaptada para unos gustos y necesidades. Y eso también es libertad.

http://www.hedgewars.org/



Open Raider

Muchos juegos libres incorporan el nombre Open delante para anunciar el tipo de licencia. Los que son clones de otros juegos comerciales de la actualidad o que han tenido éxito en el pasado, son de los más populares. Open Raider es el clon libre de la saga Tomb Raider que tanto éxito tuvo a finales de la década de los noventa y principios de ésta. Tal éxito tuvo Lara Croft, y el juego en sí, que fue llevado a la gran pantalla.

Su secuela libre, está todavía en una fase de desarrollo algo verde, podríamos decir que jugar literalmente no podemos aún, pero si podemos recorrer los escenarios con la protagonista y comprobar ciertos detalles. A pesar de todo, hay que reconocer el enorme esfuerzo que realiza la comunidad de desarrolladores de este proyecto, porque un software de este tipo: gráficos tridimensionales, escenarios complejos y de gran tamaño, diferentes armas, etc. tiene un coste de desarrollo en tiempo realmente alto. La última versión estable liberada es la 0.1



Figura 4. Openraider

y la 0.1.1 está disponible para descargar, pero todavía está en fase de desarrollo.

El juego no está inspirado en ningún capítulo concreto de la saga, sino que está más o menos inspirado en todos a la vez. La calidad gráfica no está nada mal, teniendo en cuenta que todavía nos encontramos en una versión muy preliminar. Por supuesto no es un juego comercial pero no tiene, ni tendrá, mucho que envidiarles a muchos títulos en un futuro. Respecto a sistemas operativos donde podéis probarlo, deciros que sólo podréis hacerlo en GNU/Linux, al menos de manera directa, no se si será posible compilar el código fuente con alguna dificultad en otras plataformas como Microsoft Windows. Para GNU/Linux además del código fuente también tenemos binarios precompilados y ejecutables y un paquete .deb preparado para instalar.

En definitiva, una semilla más a tener en cuenta, que ilusionará a todos los fans de la famosa saga y que tiene un futuro muy prometedor. La forja en la que se desarrolla, Sourceforge, la conoceréis todos y si tenéis tiempo suficiente puede que os interese colaborar con el proyecto, porque como hemos dicho, este tipo de proyectos tiene un largo tiempo de desarrollo. Espero que pronto, podamos volver a hablar de este título anunciándoos que ya está disponible la versión 1.0.

http://openraider.sourceforge.net/



Página de LiNUX+ está cambiando

Nuestro objetivo es transformarla en un sitio donde podrás encontrar toda la información sobre GNU/Linux y Software Libre que necesites



¡Bienvenido a nuestra página de incio!

Aquí leerás todo sobre el número actual de LiNUX+, tomarás parte en la encuesta o conocerás las conferencias que patrocinamos



En la sección de descarga encontrarás artículos y revistas que podrás leer online o descargar en pdf. Cada dos semanas añadimos un artículo nuevo y cada treinta días: un número entero de LiNUX+

¿Quieres más? En el centro de descarga te ofrecemos una lista de programas libres que te pueden ser muy útiles en el trabajo diario con GNU/Linux

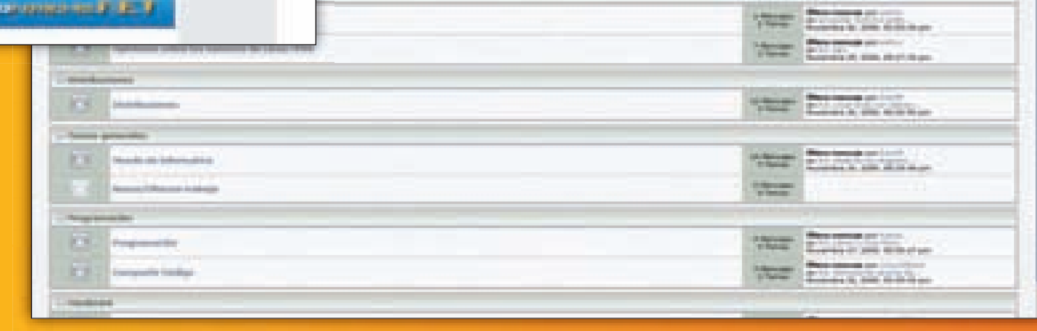




LiNUX+ patrocina varios eventos de Software Libre en España y América Latina. Echa un vistazo si quieres estar al tanto con los eventos más grandes y más famosos



Somos amigos de muchos portales y blogs dedicados a GNU/Linux y Software Libre. ¿Quieres saber más? Entra en la pestaña de Web Amigas



Recientemente hemos iniciado un foro para nuestros lectores, betatesters, autores y todas las personas interesadas en el tema de GNU/Linux. Comparte tu opinión no solo con nosotros, sino con todos a los que les fascina la misma cosa: el mundo libre

Interceptando
conversaciones Messenger

David Puente Castro

Analizar hasta qué punto un protocolo puede considerarse como vulnerable, es algo que conlleva un gran esfuerzo, pero muy por encima podemos agruparlos en dos grupos: Los que ofrecen cifrado, y los que no. Aquellos que no lo hacen, deben de atenerse a las consecuencias.



linux@software.com.pl

¿A quien le puede interesar capturar conversaciones? No encontrar respuesta a esta pregunta resulta francamente complicado. Que sea o no éticamente correcto es un tema tratado ya en demasía y que depende únicamente de los fines de la persona que lo realiza.

Capturar conversaciones realizadas a través del software *Messenger*, *aMSN*, o todo que trabaje bajo el mismo protocolo, puede ser algo interesante. Y aquí no nos referimos a la acción en si, sino al ambiciado: ¿Cómo hacerlo?.

Los programas son criaturas extrañas, la mayoría de las veces complejas y algunas veces son obras demasiado personales. Cuando uno deja de ser un simple novato y comienza a interesarse en cómo funciona lo que le rodea, e incluso decide meter su cabeza en el código fuente, no todo es color de rosa.

Y todo ello no es porque *el Programador* sea una entidad retorcida cuya única intención sea hacer su código indescifrable (aunque podría serlo, y más bonito todavía sería descifrarlo). La razón es que cuando uno consigue una base lo suficientemente sólida como para mantenerse por si misma, el programador no puede evitar la incesante necesidad de añadir nuevas funciones, introducir nuevas variables, y reestructurar el código hasta tal punto que la idea original para el que fue creado se torne ilegible (a pesar de que siga funcionando igual o mejor que antes).

¿Que pretende este artículo?

En una primera fase veremos cómo capturar conversaciones de *Messenger* con menos de 200 líneas de código fuente en lenguaje C (no contabilizando comentarios). El método que utilizaremos puede ser reutilizado para capturar conversaciones de IRC, AIM, Yahoo!, ICQ e incluso para capturar direcciones URL en peticiones HTTP.

Finalmente estudiaremos cómo este problema de seguridad puede ser evadido con la aplicación de ciertas medidas de cifrado de protocolo gracias al uso SIMP.

Ideas

Plantar un *sniffer* en su propio ordenador para capturar conversaciones puede parecer inútil, no obstante, cuando lo sitúa en los PC's de sus empleados y espera a saber a qué se dedican realmente en horario de trabajo, el asunto se vuelve más interesante. Debo aclarar que espiar a su pareja es más inmoral todavía.



La mayor ventaja se obtiene cuando entra en juego un ataque *Man In The Middle*, entonces todo cambia, pero la situación se vuelve ilícita, téngalo en cuenta.

Aplicación de monitorización y análisis de tráfico de una red que detecta en los paquetes individuales pratones específicos seleccionados por el usuario. Pueden ser utilizados ilegalmente para interceptar tráfico y por norma general suelen ser silenciosos.

Man In The Middle

Un ataque *Hombre en el medio*, es aquel en que una persona con un dispositivo de comunicación consigue situarse en una posición intermedia entre un router y el usuario legítimo de una red, de modo que todo el tráfico que circula por la red pasa a través del atacante sin el conocimiento directo del resto de participantes (ver Figura 1).

Con este artículo no se pretende incurrir en ninguna infracción de la ley, tan solo es un estudio detallado de la situación a la que nos enfrentamos a diario. El objetivo es conocer hasta qué punto estamos a merced de atacantes con malas intenciones y cómo defendernos.

Requisitos

Los requisitos indispensables son los siguientes:

- LibPcap (version 0.8) (*http://sourceforge.net/projects/libpcap/*),
- LibNet (version 1.1 o superior) (*http://libnet.sourceforge.net)*.

LibPcap es una pequeña maravilla que nos proporciona todas las funciones necesarias para capturar paquetes y controlar la información en ellos contenida. LibNet, es todo lo contrario, su objetivo es facilitar un API de programación orientado a la inyección de paquetes.

No se entrará en detalles acerca de la implementación y argumentos de las funciones de la librería *libpcap*, sin embargo, se mostrará el cometido de cada una de ellas, pues para eso estamos aquí, para comentar el código. *Libnet*, por su parte, podría ser evitado pero utilizaremos sus estructuras de cabecera para protocolos debido a que son verdaderamente intuitivas y facilitan nuestra tarea.

Análisis del protocolo MSN

El método utilizado para crear el código que veremos en la siguiente sección se basa en algo tan sencillo como interpretar las capturas de uno de los mejores *sniffer* de redes que disponemos desde nuestro Sistema Operativo Linux: *Wireshark*.

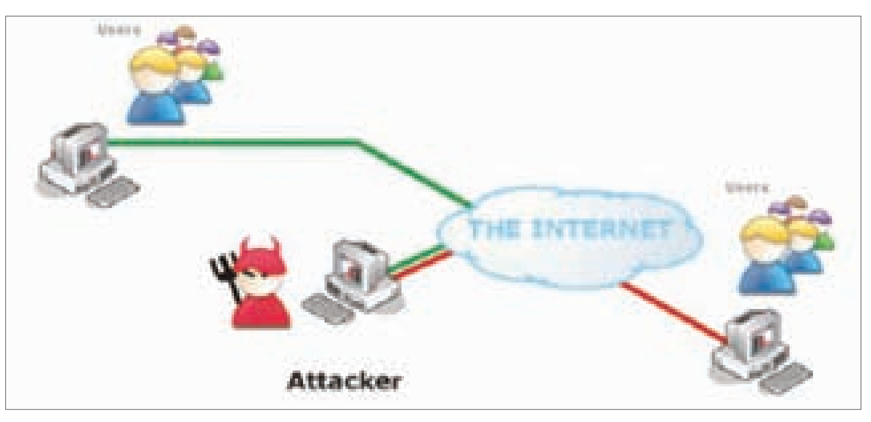


Figura 1. Esquema del ataque Man in the Middle

Anteriormente conocido como *Ethereal*, es la herramienta más destacada en Linux para el estudio de lo que sucede en las capas más bajas de nuestras conexiones de red. Su principal capacidad es la de capturar todos los paquetes que entran y salen de nuestra red, y generar una estructura muy intuitiva con toda la información legible que una persona con ciertos conocimientos puede interpretar. Lo mejor que se puede decir de él, es que es de código abierto.

Podríamos decir y decimos que el protocolo Messenger, al igual que por ejemplo IRC y FTP tal vez, funciona por medio de comandos. Estos son dirigidos a un servidor *central* que se encarga de hacer las gestiones necesarias según los parámetros indicados. Este servidor es un intermediario entre emisor y receptor. En realidad todo el proceso es un poco más complicado, pero esta idea sirve perfectamente a nuestros fines.

Nota: Ahora deténgase a pensar un momento. Si nuestra información viaja sin cifrar y circula por una máquina que no es de nuestra confianza, ¿hasta qué punto está seguro de que sus mensajes no son inspeccionados en busca de información peligrosa?

Bien, continuemos. Lo que hemos dicho del servidor central y de la interactuación a través de comandos es tan cierto, que puede establecer una comunicación con toda naturalidad mediante una clásica sesión de *telnet*. Puede seguir este estudio aquí (*http://65.23.158.196/nitz/nitz/protocolo_msn.pdf*).

Describamos ahora los comandos básicos, para posteriormente centrarnos en el que más interesa en nuestra investigación:

- *VER*: Establece la versión del protocolo Messenger.
- *USR*: Comando de identificación de usuario.
- *SYN*: Obtener la lista de contactos.
- *CHG*: Comando de *cambio de estado*.
- *CAL*: Invitar a un contacto a una conversación.
- *BYE*: Comando de cierre de sesión.

Existen algunos otros comandos, pero están orientados a control y establecimiento de parámetros de conexión. Ha llegado el momento de que veamos el comando más importante en nuestra investigación.

Comando MSG

Veámoslo en directo. Inicie Wireshark con permisos de administrador `root`. Una vez se encuentre delante de la pantalla principal siga el menú *Capture -> Options*. Una vez aquí escogeremos como interfaz de red aquella con la que obtenemos nuestra conexión normal a internet (en mi caso `ath0`) y en el campo *Capture Filter* escribiremos lo siguiente:

```
tcp and src port 1863 or dst port 1863
```

Si hacemos click en el botón *Start* estaremos iniciando una captura de red filtrando únicamente el trafico TCP que entra y sale del puerto 1863, que precisamente es el que Messenger utiliza para las comunicaciones.

Inicie su software *aMSN* habitual en Linux, y comience una conversación. Después de un breve intercambio de frases puede cerrar la sesión y detener Wireshark. Veamos qué hemos obtenido en nuestro ejemplo (Figura 3).

Esto es un inicio de sesión clásico. En el marco superior, hacia la derecha, podemos



Figura 2. Logo de Wireshark

Listado 1a. Código fuente

```
/* Archivos de cabecera y definiciones */

 #include <stdio.h>
 #include <pcap.h>
 #include <libnet.h>

 #define MSN_PORT 1863
//  Puerto por defecto para MSN Messenger.
 #define TCPHDR_LEN 0x20
//  Algunas veces podría necesitar cambiar esto
//  por "0x14" para que funcione. Ello es debido
//  a que diversas opciones del protocolo TCP
//  pueden cambiar el tamaño de la cabecera de
//  20 a 32 bytes .
#define FILTRO_MSN "tcp and src port 1863 or dst port
1863"
// Declaraciones de funciones
 static char * get_ident(char *);
 void print_msg(char *, int, char *, char *);
 void handle_msg(char *, char, int);
 void read_msn(u_char *, const struct pcap_pkthdr *,
const u_char *);
 int main(int argc, char *argv[])
 {
   int rc;
   char *device;
   char errbuf[PCAP_ERRBUF_SIZE];
   struct bpf_program filtro;
   bpf_u_int32 netp, maskp;
   pcap_t* sniffmsn;
   if (argc < 2)
      exit(0);
   device = argv[1]; // Único parámetro: interfaz de
red. Ex.: eth1, ath0, etc.
// Abrimos el dispositivo para captura en modo
promiscuo
   sniffmsn = pcap_open_live(device, 1600, 1, 20,
errbuf);
   if (sniffmsn == NULL) {
      fprintf(stderr, "pcap_open_live(): %s\n",errbuf);
      exit(-1);
   }
   if (pcap_lookupnet(device, &netp, &maskp, errbuf) ==
-1) {
      fprintf(stderr, "Error en pcap_lookupnet(): %s\
n", errbuf);
      exit(-1);
   }
// Creamos el filtro con las opciones anteriormente
definidas
   if (pcap_compile(sniffmsn, &filtro, FILTRO_MSN, 0,
netp) == -1) {
      fprintf(stderr, "Error compilando el filtro\n");
      exit(-1);
   }
// Aplicamos el filtro a la interfaz
if (pcap_setfilter(sniffmsn, &filtro) == -1) {
      fprintf(stderr, "Error aplicando el filtro\n");
      exit(-1);
   }
// Iniciamos la captura de paquetes indefinidamente, el
3er parámetro es la
// función que se encarga de interpretar los paquetes,
siempre tiene tres
// parámetros y no tiene valor de retorno.
   rc = pcap_loop(sniffmsn, -1, &read_msn, NULL);
   return 0;
 }
// Función que interpreta los paquetes. Extraemos las
cabeceras TCP e IP y
// controlamos el payload (carga o datos) que se pasará
a la siguiente
// función diferenciando entre enviados y recibidos
para leer los que sean
// realmente interesantes.
 void
 read_msn(u_char *useless, const struct pcap_pkthdr
*pkthdr, const u_char *pkt)
 {
   u_char *data;
   int len;
   struct libnet_ipv4_hdr *iph; // Cabecera IP
   struct libnet_tcp_hdr *tcph; // Cabecera TCP
   iph = (struct libnet_ipv4_hdr *)(pkt + LIBNET_ETH_H);
   tcph = (struct libnet_tcp_hdr *)(pkt + LIBNET_ETH_H
+ LIBNET_IPV4_H);
// Los datos se alcanzan tras pasar las cabeceras:
ethernet, ip y tcp.
   data = (u_char *)(pkt + LIBNET_ETH_H + LIBNET_IPV4_H
+ TCPHDR_LEN);
   len = ntohs(iph->ip_len);
// Si el puerto origen es 1863, estamos recibiendo un
mensaje.
   if (ntohs(tcph->th_sport) == MSN_PORT) {
      handle_msg(data, 'R', len);
// Maneja los datos que anteriormente
// destripamos: comando, opciones y mensaje.
   }
// Si el puerto destino es 1863, estamos enviando un
mensaje
   else if (ntohs(tcph->th_dport) == MSN_PORT) {
      handle_msg(data, 'S', len);
   }
 // Función que busca mensajes dentro de la carga útil
del paquete.
 void
 handle_msg(char *data, char dir, int dlen)
 {
   char *pc, *pstart;
   char *email;
   char *nick;
   char *buf;
```



Listado 1b. Código fuente

```
// Creamos un buffer con la longitud del payload
   buf = (char *) calloc(dlen, sizeof(char));
// Copiamos allí su contenido para manejarlo
   if (buf != NULL) {
      strncpy(buf, data, dlen);
   } else {
      fprintf(stderr, "\nNo hay suficiente memoria\n");
      exit(-1);
   }
// Comprobamos que contenga el comando "MSG"
   if (strncmp(buf, "MSG", 3) == 0) {
// Que su contenido sea texto plano y no datos de
control
      if (strstr(buf, "Content-Type: text/plain") !=
NULL) {
// Mensajes enviados
         if (dir == 'S') {
// Nos situamos en el primer parámetro del comando MSG
            pc = strchr(buf + 4, ' ');
            pc++;
// Función que alcanza el mensaje y lo imprime. El
último parámetro
// es nulo porque no imprimimos el e-mail del emisor,
nosotros.
// Deberíamos hacerlo si utilizamos ataques MITM.
            print_msg(pc, dir, NULL, NULL);
         }
// Mensajes recibidos
         else {
// Lo mismo que antes pero esta vez colocamos un
caracter nulo al
// final de la dirección e-mail del receptor para poder
manejar
// este fragmento como una cadena.
            pstart = buf + 4;
            pc = strchr(pstart, ' ');
            if (pc != NULL)
               *pc = 0;
            email = get_ident(pstart);
// Esta función, que será descrita al
// final del código, no es más que un
// pequeño administrador de memoria
// dinámica, que reserva el espacio
// suficiente para almacenar e-mail y
// sus consecuentes overflows.
// Mismo procedimiento para almacenar el nick.
            pc++;
            pstart = pc;
            pc = strchr(pstart, ' ');
            if (pc != NULL)
               *pc = 0;
            nick = get_ident(pstart);
            pc++;
            print_msg(pc, dir, email, nick);
// Imprimir mensaje
         }
      }
   }
// Trabajar limpiamente
   free(buf);
   free(email);
   free(nick);
}
 void
print_msg(char *pc, int dir, char *mail, char *nick)
{
 char *str, *str_end;
 int len;
// Parametros válidos para el comando "MSG"
 if (*pc == 'U' || *pc == 'N' || *pc == 'A') {
   str = strchr(pc, ' ');
   str++;
 }
 else {
   str = pc;
 }
// Justo antes del primer retorno de carro se
encuentra la longitud del
// mensaje, lo almacenamos en la variable 'len'.
 str_end = strchr(str, '\r');
 *str_end = '\0'; // Ya saben, para manejar cadenas
deben terminar en \0.
 len = atoi(str);
 str = str_end + 2;
 *(str + len) = '\0';
// Gracias al retorno de carro y nueva linea adicional
que el protocolo
// MSN nos facilita, el mensaje real siempre se
encontrara después de dos
// retornos de carro y nueva linea consecutivos.
 str = strstr(str, "\r\n\r\n");
 str += 4;
 if (dir == 'S')
 else
    printf("\nSMS desde (%s) [%s]: %s\n", mail, nick,
str); // Recibidos
}
// Administrador de memoria dinámica para almacenar
cadenas.
static char *
get_ident(char *ptr)
{
   char *buff;
   size_t bsize = 32; // Tamaño de buffer inicial
   int lenp = strlen(ptr);
// Mientras la longitud de la cadena sea mayor que el
tamaño de buffer,
// lo vamos multiplicando por 2 hasta que la capacidad
sea suficiente.
   while (lenp > bsize) {
      bsize *= 2;
   }
   buff = (char *) calloc(bsize, sizeof(char));
// Creamos el buffer con 0's
```

Listado 1c. Código fuente

```
    if (buff != NULL) {
        strncpy(buff, ptr, bsize
- 1);
// Copiamos la cadena al buffer
        return buff;
    } else {
        fprintf(stderr, "\nNo hay
suficiente memoria\n");
        exit(-1);
    }
 }
```

ver remarcado un comando USR con ciertos parámetros de identificación para mi usuario. Encima verá algunos referentes a la versión. Todo esto es más o menos común a cada implementación, pero veamos ahora qué comandos MSG son relevantes y cuáles no (Figura 4).

Sólo las líneas 23 y 33 son relevantes. Podemos ver varios comandos MSG, pero no todos son de interés y no todos contienen mensajes. Echemos un vistazo al marco inferior de nuestro paquete número 23 que hemos seleccionado intencionadamente. Al final de todo se observa el mensaje que hemos enviado (aquí, *hola, mi nombre es David*), pero se preguntará que es toda esa información previa. Una breve descripción:



Figura 5. Redireccionando los mensajes hacia un archivo de salida

- MSG 58 A 146 \rn:
  - Nombre del comando,
  - Parámetro no importante,
  - Debemos saber que solo “A”, “N” y “U” sirven a nuestros intereses,
  - Longitud en bytes de todo lo que viene a continuación de este parámetro hasta el final del mensaje enviado.
- MIME-Version: 1.0\rn:
  - Por defecto.
- Content-Type: text/plain; charset=UTF-8\rn:
  - El parámetro `text/plain` indica que el paquete lleva texto legible.Este es uno de los puntos más relevantes, solo los paquetes que contengan este campo llevarán mensajes de conversación consigo. Los demás podemos desecharlos.
  - Formato de codificación de caracteres.
- X-MMS-IM-Format: FN=Helvetica; EF=; CO=000000; CS=0; PF=22\rn:
  - Fuente de la letra,
  - Cursiva, Negrita o Subrayado,
  - Color del texto,
  - Codificación de caracteres,
  - ...
- \rn:
  - Adicional, será de mucha ayuda en la codificación del programa,
- hola, mi nombre es David:
  - ¿Requiere alguna explicación?

Figura 3. Ilustración al comando MSG

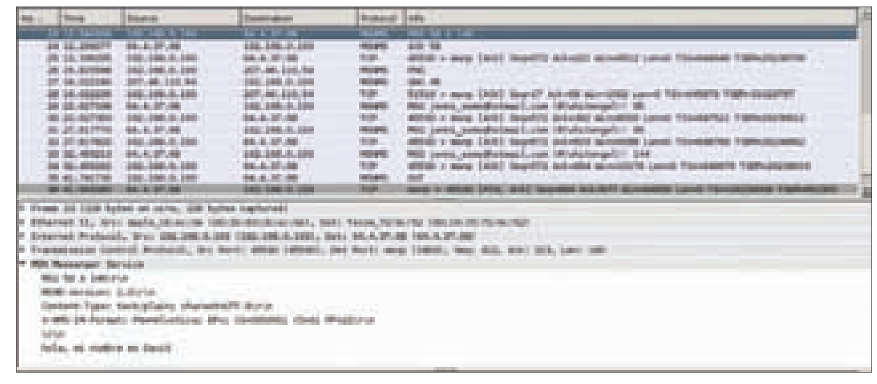


Figura 4. Comprobando la relevación de comandos MSG

User-Agent que identifique el software desde el que se está realizando la conexión, como ocurre con los navegadores.



Hasta aquí todo claro, lo que acabamos de ver es el mensaje que nosotros enviamos hacia el destinatario. Entonces las ideas que sacamos en claro de todo el esquema son las siguientes:

- Solo los comandos MSG con parámetros “A”, “N” o “U” son atractivos,
- Solo paquetes con la cadena `Content-Type: text/plain;` contienen mensajes,
- El mensaje siempre se encuentra después de una secuencia `\r\n\r\n`.

Si ahora miramos el contenido del paquete número 33, encontramos la única diferencia:

- MSG jenni_somo@hotmail.com (#)whitengel!! 144\rn:
  - Nombre del comando,
  - Dirección de correo del usuario,
  - Nick o frase del usuario,
  - Cantidad de bytes hasta el final del mensaje.

Casi sin darnos cuenta acabamos de vislumbrar como nuestra privacidad diaria se encuentra totalmente comprometida y en manos de cualquiera que tenga la capacidad de aprovecharse de la situación.

Paso a paso

Para la realización de nuestro programa seguiremos una estructura *paso a paso* que facilitará en cierta medida su comprensión a la hora de interpretarlo. Veámoslo:

- Capturar paquetes con libpcap.
- Filtrar los que vayan o provengan del puerto 1863 (msn).
- Copiar la carga de datos del paquete a un buffer.
- Diferenciar entre enviados y recibidos.
- Guardar el e-mail/nick del receptor en caso necesario.
- Seleccionar aquellos que contengan el comando `MSG`.
- Seleccionar aquellos que contengan `Content-Type: text/plain`.
- Desplazarnos hasta la cadena del mensaje e imprimirla (tras “\rn\rn”).

Aclaraciones

Puede compilar el código fuente (Listado 1) de esta forma:

```
$ gcc smsn.c -lpcap -lnet -o smsn
```

Habíamos mencionado al principio del artículo que un *sniffer* debe de ser un programa silencioso. Y esto debe cumplirse tanto en el hecho de no interferir en el tráfico que circula por la red, como en el hecho de que no sea visible localmente (si estuviésemos trabajando bajo un ataque *Man in The Middle* esto no sería necesario puesto que el *sniffer* se activa en el ordenador del atacante y no en las víctimas).

No es que sea muy ortodoxo, pero para no complicar más el código que acabamos de ver hace un momento, tomaremos de la mano las facilidades que nos aporta Linux en la línea de comandos. Los objetivos son, ocultar el *sniffer* visualmente y redireccionar los mensajes hacia un archivo de salida que hará las veces de *log* o archivo de grabación. Esto puede conseguirse fácilmente con la siguiente instrucción:

```
# nohup ./smsn [dev] > log_msn.log &
```

o

```
$ sudo nohup ./smsn [dev] > log_
msn.log &
```

Con `nohup` usted desengancha el siguiente comando del shell y así puede cerrarlo sin preocuparse. El carácter final “&” envía el proceso al segundo plano. Solo nos queda verlo en acción (Figura 5).

Usted no es como los demás (SIMP)

Hemos visto que es posible capturar conversaciones realizadas por medio del protocolo MSN. Hemos visto que es posible crear una pequeña utilería como prueba de concepto. Pero todavía no hemos visto que es posible defenderse.

SIMP es capaz de proteger el contenido de nuestras conversaciones mediante algoritmos de encriptación. El programa trabaja clasificando las conversaciones según 3 niveles de seguridad: No encriptado, encriptado, y encriptado/autentificado.

SIMP fue principalmente diseñado en forma de una aplicación llamada SimpLite para entornos Windows, que dispone de cuatro versiones distintas según nuestro deseo sea cifrar una de estos servicios de mensajería instantánea:

- MSN,
- Yahoo!,
- ICQ/AIM,
- Jabber/Google.

Afortunadamente para nosotros, la empresa SecWay diseñó una versión para Unix, SimpServer, que puede correr en los siguientes sistemas:

- Linux x86 (libstdc++ 5 required),
- MacOS X,
- FreeBSD 5.



Figura 6. Opciones avanzadas



Sobre el autor

David Puente Castro, alias *blackngel*, es un gran aficionado a la programación y la seguridad informática . Original de Ourense (Galicia) y viviendo actualmente en la provincia de Salamanca dedica la mayor parte de su tiempo libre a disfrutar de la experiencia Linux.

Asiduo escritor de artículos básicos sobre temas de seguridad informática en el E-Zine electrónico S.E.T. (Saqueadores Edición Técnica), ampliamente conocido en el Underground Hispano. Actualmente mantiene su segunda página oficial en: *http://set.diazr.com/*, cuyo principal objetivo, radica en ocupar el lugar que en su momento cubrió la web de *Hackemate* como repositorio de material sobre temas de *hacking*.

Su primer encuentro con Linux data ya de hace unos 8 años y fue poco más adelante que descubrió el fantástico mundo de la programación y la subcultura hacker. Participa activamente en *wargames* como: *yoire.com*, *warzone* (*elhacker.net*) y otros.

Aunque trabaja con ordenadores, la mayor parte de sus tareas diarias no tienen una relación directa con esta temática. Auxiliar administrativo, obligado a utilizar el sistema operativo Windows, tiene la costumbre de llevarse su última *Distribución Live* a la oficina para seguir disfrutando de su SO favorito, Linux, y desarrollar sus pequeños proyectos.

Dedica el artículo a su pareja y familia por la paciencia que demuestran cuando intenta explicarles á haciendo.

Por lo demás, puede encontrarlo online, prácticamente las 24 horas del día, en las siguientes direcciones: *blackngel1@gmail.com* y *black@set-ezine.org*



Figura 7. Página web de LibPcap

Es muy importante notar que la comunicación es cifrada punto a punto, y la persona con la que deseemos establecer una comunicación también debe tener instalada esta misma herramienta, de otro modo los mensajes seguirían viajando en texto claro.

SimpServer es un servicio que aprenderemos a instalar y configurar a continuación (lo haremos bajo un sistema Ubuntu Linux):

- Descargar el programa desde esta dirección: *http://download.secway.com/public/products/simpserver/simpserver-2.1.5c-linux-x86.tgz*
- Instalar librería dependiente: `$ sudo apt-get install libstdc++5`
- Descomprimir en `/usr/local`:
  `$ cp simpserver-2.1.5c-linux-x86.tgz /usr/local`
  `$ tar -xvzf simpserver-2.1.5c-linux-x86.tgz`
- Ejecutar el servicio (Listado 2).

SimpServer trabaja como un proxy, utilizando SOCKS 4 para tal fin. Por ello debemos configurar nuestro cliente de mensajería para utilizarlo. Por desgracia aMSN solo soporta SOCKS 5, pero los usuarios de Linux siempre tenemos una solución para todo y en este caso el cliente Pidgin nos viene de perlas. Dando por supuesto que sabe como instalarlo en su sistema (`$ sudo apt-get isntall pidgin`), veamos como configurarlo (Figura 6):

- *Cuentas->Gestionar Cuentas->Modificar ->Avanzadas,*
- Tipo de proxy: SOCKS 4,
- Servidor: localhost,
- Puerto: 11863.

Hasta aquí tiene configurado SimpServer totalmente en su sistema y disponible para ser utilizado con Pidgin. Cualquier futura comunicación que implique a dos usuarios con este software funcionando (SimpServer), producirá un mensaje informando acerca de un intercambio de claves. Claves que han sido generadas tanto de forma automática como de forma manual en anteriores versiones.

Tiene deberes, compruebe ahora con Wireshark en que formato fluyen los datos cuando atraviesan la red. Ya puede sentirse un poco más seguro.

Conclusión

Aunque no lo parezca, una de las partes más interesantes del código que hemos desarrollado es el pequeño administrador de memoria `get_ident(...)`. Con él podemos aprender prácticas de programación bastante más adecuadas en la actualidad. Debemos evitar caer en las tentaciones del

Programación en Linux: Casos Prácticos (*http://www.anayamultimedia.es*), podría ser una buena lectura si pretendes que tu visión ante la programación cambie de forma notable.

Espero que esto haya sido lo suficientemente interesante como para llegar hasta el final. La idea que subyace a todo este asunto radica en que debemos ser precavidos con todo aquello en lo que confiamos.

En todas las actividades es saludable, de vez en cuando, poner un signo de interrogación sobre aquellas cosas que por mucho tiempo se han dado como seguras. [Bertrand Russell].

Hasta la próxima, y feliz hacking.

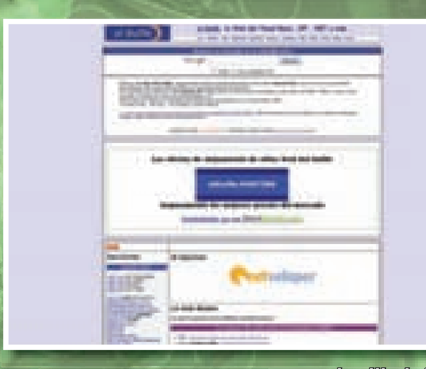


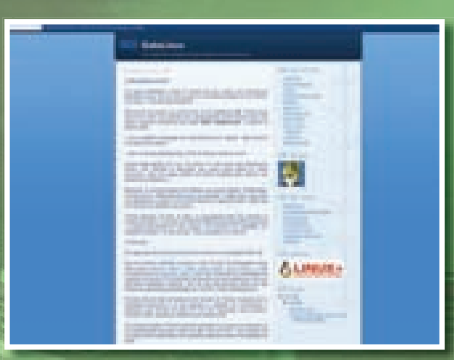
En la red

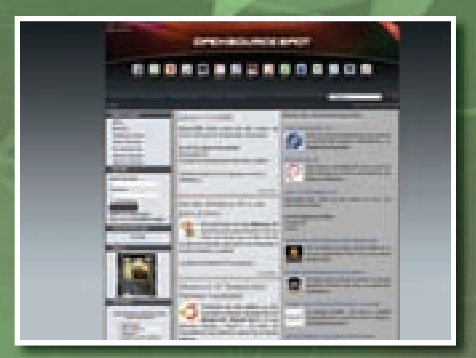
- LibPcap:
  http://sourceforge.net/projects/libpcap/
- LibNet:
  http://libnet.sourceforge.net
- Protocolo MSN:
  http://65.23.158.196/nitz/nitz/protocolo_msn.pdf
- Programación en Linux: Casos Prácticos – ISBN: 978-84-415-1839-1:
  http://www.anayamultimedia.es
- SimpServer para Unix:
  http://www.secway.fr/us/products/simpserver/

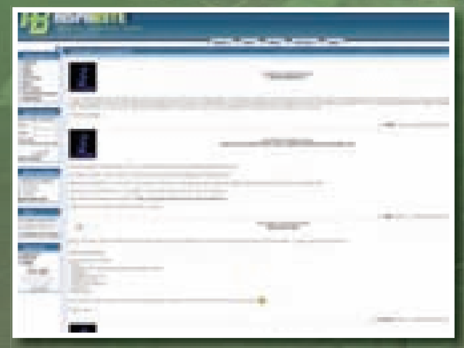
Páginas recomendadas

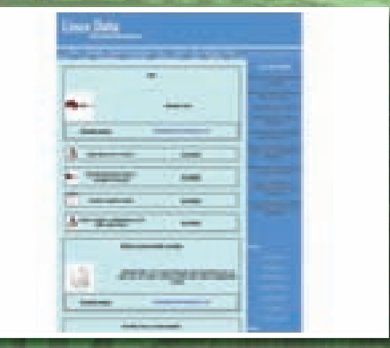

www.diariolinux.com


www.elguille.info


www.gatolinux.blogspot.com


www.opensourcespot.org

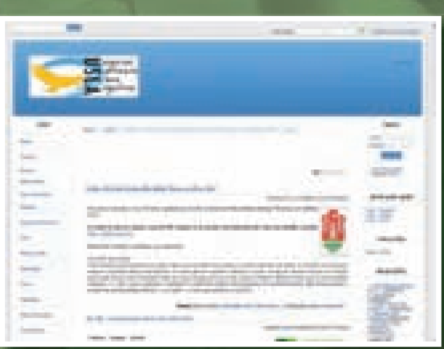

www.hispabyte.net

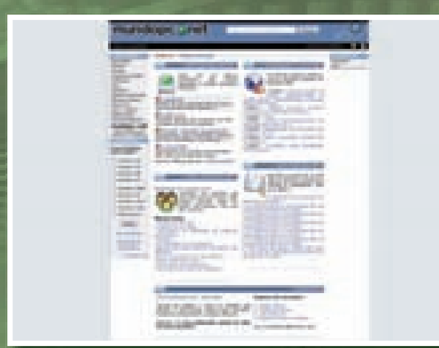

www.linuxdata.com.ar


www.linuxhispano.net

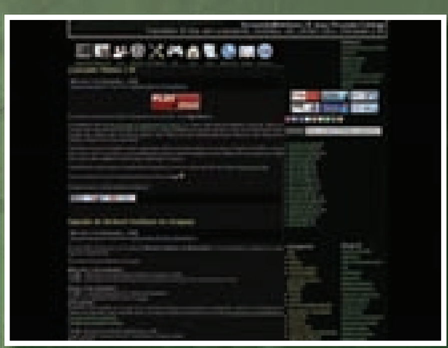


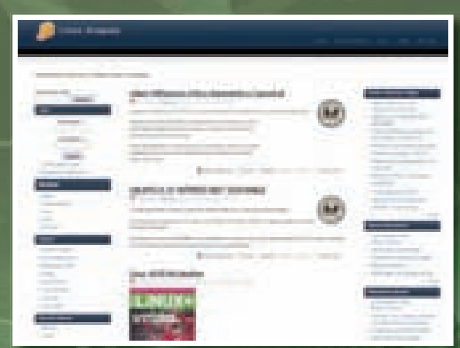
www.pillateunlinux.wordpress.com


www.usla.org.ar



www.mundopc.net


www.picandocodigo.net


www.linuxuruguay.org

Firewall
con 2 canales de internet

Samuel Diaz

El presente documento nace de la necesidad que surgió de poder configurar un acceso a Internet a través de 2 canales, utilizándolos simultáneamente. Todo esto debe lograrse sin modificar la estructura de red actual, o conservando el acceso a todos los recursos que los usuarios tienen antes de esta implementación.
Con esta implementación no se restringe a únicamente 2 canales, se puede llegar a realizar con más canales de Internet.



linux@software.com.pl

El acceso a Internet ya no debe depender de un único proveedor, para estar presente en la red se debe asegurar que tanto la infraestructura tecnológica como los canales de comunicaciones se encuentren disponibles en todo momento. Al surgir esta necesidad y no encontrar un documento que me pudiera ayudar a solucionar esta problemática en conjunto, me di a la tarea de plasmar mis experiencias en la solución de esta problemática.

Para atacar este problema se debe dividir es 2 partes:

- Seguridad,
- Enrutamiento.

La parte de seguridad se trabaja con iptables, para crear reglas de filtrado de paquetes que permitan la salida hacia Internet de los usuarios por un canal o por el otro, según sean creadas estas reglas (reglas de NAT).

La parte de enrutamiento se trabaja con un proyecto llamado iproute2 *http://linux-net.osdl.org/index.php/Iproute2* el cual nos permite realizar la tarea de enrutamiento por bien sea por un canal o por el otro. Este proyecto como iptables ya se encuentra disponible en una gran mayoría de distribuciones.

Para que todo en conjunto funcione correctamente la parte de seguridad y el enrutamiento deben ir de la mano con las mismas configuraciones y los mismos parámetros.

Procedimiento para configurar un firewall en linux con accesso a dos canales de internet

El diagrama de red presentado se compone básicamente de una red DMZ con direccionamiento *10.10.10.0/24*, cuyo default



Lo que deberías saber...

Un cortafuegos no puede protegarse contra ataques desde du punto de operación.



Lo que vas a aprender...

- Usar un dispositivo de seguridad atraves de 2 canales.
- Como configurar firewall, como protección de informacion privada en Linux.



Listado 1. Primer canal de internet (default gateway es 192.168.1.1.)

```
#!/bin/bash
# Samuel Diaz
# Enero de 2007
# Bogota - Colombia
#Tabla de Rutas Red de pruebas
# Se limpian las tablas de enrutamiento y se reinician
las interfaces de red para limpiar todas las posibles
rutas que se tengasn
echo "Limpiando Tablas de enrutamiento"
ip route flush all
ip route flush table 201
ip route flush table 202
ip route flush table 222
ip rule show | grep -Ev '^(0|32766|32767):' \
      | while read PRIO RULE; do
      ip rule del prio ${PRIO%%:*} $( echo $RULE | sed
's|all|0/0|' )
      ip rule del prio ${PRIO%%:*} $( echo $RULE )
done
ip route flush default
rcnetwork restart
#Definicion de interfaces de red, Redes locales, DMZ y
redes de canales de Internet.
echo "Configuracion Automatica de Rutas"
# Interface eth0, Primer canal
IF0="eth0"
# Interface eth1, red local
IF1="eth1"
# Interface eth2, DMZ
IF2="eth2"
# Interface eth3, segundo canal
IF3="eth3"
# IP eth0, primer canal
IP0="192.168.1.70"
# IP eth1, red local

# IP eth2, DMZ
IP2="10.10.10.70"
# IP eth3, segundo canal
IP3="192.168.9.70"
# Gateway Primer Canal de Internet
P0="192.168.1.1"
# Gateway Segundo Canal de Internet
P3="192.168.9.111"
# Red primer canal
P0_NET="192.168.1.0/24"
# Red local
P1_NET="172.16.1.0/24"
# Red DMZ
P2_NET="10.10.10.0/24"
# Red Segundo Canal
P3_NET="192.168.9.0/24"
# Definicion de regionales
#Medellin
MEDELLIN="172.17.1.0/24"
#Cali
CALI="172.18.1.0/24"
# Se definen las rutas de las redes donde se encuentran
las
# salidas de internet, junto con el gateway de cada
# canal y se asocian a las tablas T1 y T2.
ip route add $P0_NET dev $IF0 src $IP0 table T1
ip route add default via $P0 table T1
ip route add $P3_NET dev $IF3 src $IP3 table T2
ip route add default via $P3 table T2
ip route add $P0_NET dev $IF0 table T1
ip route add $P3_NET dev $IF3 table T1
ip route add $P2_NET dev $IF2 table T1
ip route add 127.0.0.0/8 dev lo table T1
ip route add $P0_NET dev $IF0 table T2
ip route add $P3_NET dev $IF3 table T2
ip route add $P1_NET dev $IF1 table T2
ip route add 127.0.0.0/8 dev lo table T2
#Se asocian la red local y DMZ para que se tenga
# enrutamiento en las 2 tablas.
ip route add $P2_NET dev eth2 table T1
ip route add $P2_NET dev eth2 table T2
ip route add $P1_NET dev eth1 table T2
ip route add $P1_NET dev eth1 table T1
ip route add $MEDELLIN via 172.16.1.1 dev eth1 table T1
ip route add $MEDELLIN via 172.16.1.1 dev eth1 table T2
ip route add $CALI via 172.16.1.1 dev eth1 table T1
ip route add $CALI via 172.16.1.1 dev eth1 table T2
#se define el default gateway del firewall
ip route add default via $P0
ip rule add from $IP0 table T1
ip rule add from $IP3 table T2
# Se crean las definiciones de los gateway de cada tabla,
# junto con que ip utilizaran esas tablas
ip rule add from 10.10.10.1/32 table T1
ip rule add from 10.10.10.15/32 table T2
ip rule add from 10.10.10.16/32 table T1
#ip rule add from 10.10.10.50 table T2
#ip rule add from 172.16.1.20 table T1
#ip rule add from 172.16.1.50 table T2
# Se refresca la tabla de cache
ip route flush cache
ip rule add from 172.16.1.0/24 table T1
ip rule add from $MEDELLIN table T1
ip route flush cache
#route add default gw 192.168.1.1
route add -net 172.17.1.0/24 gw 172.16.1.1
```

gateway es *10.10.10.70* y se encuentra conectada a la interfase eth2. Se tienen dos canales de Internet en redes diferentes, la primera se encuentra en la red *192.168.1.0/24* conectada a la interfase eth0 y con dirección IP *192.168.1.70*, para esta red se cuenta con un gateway *192.168.1.1*. El segundo canal se encuentra conectado a la interfase eth3 y con dirección IP *192.168.9.70*, este canal cuenta con el gateway *192.168.9.111*. La red LAN se encuentra conectada a la interfase eth1 y con una dirección de red *172.16.1.0/24*, el default gateway de esta

Listado 2. Segundo canal de internet (default gateway 192.168.9.111.)

```
#!/bin/sh
# ---------------------------------
# Samuel Diaz
# Enero de 2007
# Bogota - Colombia
# ---------------------------------
# ------------------------------
# modulos no implementados
# ------------------------------
modprobe ip_queue
modprobe ip_tables
modprobe ipt_MARK
modprobe ipt_REJECT
modprobe ipt_TCPMSS
modprobe ipt_TOS
modprobe ipt_tos
modprobe ipt_limit
modprobe ipt_mac
modprobe ipt_mark
modprobe ipt_multiport
modprobe ipt_owner
modprobe ipt_state
modprobe ipt_tcpmss
modprobe ipt_LOG
modprobe ipt_MASQUERADE
# --------------------------------------------------
# Soporte para ftp, IRC
# Los modulos estan en /usr/include/
# linux/netfilter_ipv4
# --------------------------------------------------
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe ip_conntrack_irc
# Funciona en kernel 2.4.19
modprobe ip_nat_ftp
#--------------------
# Habilita enrutamiento
#--------------------

#--------------------
# Habilita IP Dinamico. 1=Habilitado,
# 0=No Habilitado
#--------------------
echo "0" > /proc/sys/net/ipv4/ip_dynaddr
#--------------------
# Limpia los canales
#--------------------
iptables -F INPUT
iptables -F FORWARD
iptables -F OUTPUT
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
# ---------------------------------------------------
# La política es: todo es prohibido
# hasta que se diga
lo contrario
# Esta técnica se usa en Europa
# --------------------------------------------------
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
#------------------------------------------------
# La tarjeta de red local siempre
# debe permitir todo
#------------------------------------------------
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -i eth0 -j ACCEPT
iptables -A INPUT -i eth1 -j ACCEPT
#iptables -A INPUT -i eth2 -j ACCEPT
# ************************************************
# Permite PING para hacer pruebas
# de conectividad
# ************************************************
iptables -A INPUT -p icmp -j ACCEPT
#*****************************
****************************************
# Nat - Permite a los PC de LAN salir
# a internet
#****************************************************
****************************************************
#*** NAT - ESTAS DIR. IP
# TIENEN ACCESO A TODOS LOS SERVICIOS ***
iptables -t nat -A POSTROUTING -j SNAT -s 172.16.1.15/
32 --to 192.168.1.70
iptables -t nat -A POSTROUTING -j SNAT -s 172.16.1.20/
32 --to 192.168.1.70
iptables -t nat -A POSTROUTING -j SNAT -s 172.17.1.50/
32 --to 192.168.1.70
iptables -t nat -A POSTROUTING -j SNAT -s 10.10.10.1/
32 --to 192.168.1.70
iptables -t nat -A POSTROUTING -j SNAT -s 10.10.10.15/
32 --to 192.168.9.70
iptables -t nat -A POSTROUTING -j SNAT -s 10.10.10.16/
32 --to 192.168.1.70
iptables -t nat -I PREROUTING -p tcp -d 192.168.1.70
--dport 2525 -j DNAT --to 172.16.1.10:25
iptables -t nat -I PREROUTING -p tcp -d 192.168.9.70
--dport 25 -j DNAT --to 10.10.10.15:25
iptables -t nat -I PREROUTING -p tcp -d 192.168.9.70
--dport 21 -j DNAT --to 10.10.10.15:21
iptables -t nat -I PREROUTING -p tcp -d 192.168.1.70
--dport 21 -j DNAT --to 172.17.1.50:21
iptables -t nat -I PREROUTING -p tcp -d 192.168.1.70
--dport 25 -j DNAT --to 10.10.10.1:25
iptables -t nat -I PREROUTING -p tcp -d 192.168.1.70
--dport 80 -j DNAT --to 10.10.10.1:80
```



red es *172.16.1.70*. Desde esta red se tiene un enrutador que conecta a la red WAN con las sedes de las ciudades principales, ejemplo, Medellín, Cali, Barranquilla, etc. La dirección de este enrutador en la sede principal es *172.16.1.1*, Por ejemplo, la sede de Medellín tiene la dirección de *172.17.1.0/24* y Cali tiene *172.17.1.0/24*.

En este laboratorio se probara un balanceo de carga manual sobre los canales de Internet, para esto se situaran diferentes equipos detrás del firewall, los equipos ubicados en la DMZ tendrán la siguiente información de red:

```
Equipo1DMZ Equipo2DMZ
IP= 10.10.10.1/24 IP= 10.10.10.15/24
GW= 10.10.10.70 GW= 10.10.10.70
DNS1= 200.21.200.2 DNS1= 200.21.200.2
DNS2= 63.171.232.38 DNS2=
63.171.232.38
```

Continuando con el laboratorio, el Equipo1DMZ saldrá por el primer canal de Internet, o sea por el *gateway 192.168.1.1*, y el Equipo2DMZ saldrá por el segundo canal de Internet, o sea por el *gateway 192.168.9.111*. Estos equipos tendrán salida directa a Internet a través de reglas de NAT, lo cual les permitirá salir libremente hacia Internet. Se situara otro equipo en la red LAN con la siguiente información de red:

```
Equipo1LAN
IP= 172.16.1.50/24
GW= 172.16.1.70
DNS1= 200.21.200.2
DNS2= 63.171.232.38
```

Este equipo tendrá una regla de NAT que le permite salir libremente hacia Internet. Para complementar este laboratorio se puede situar otro equipo en la red LAN con la siguiente información de red:

```
Equipo2LAN
IP= 172.16.1.100/24
GW= 172.16.1.70
DNS1= 200.21.200.2
DNS2= 63.171.232.38
```

Este equipo no tendrá salida a Internet directamente, ya que en las reglas de firewall no se configurara una regla de NAT que le permita la salida libre a Internet. Se puede configurar un servidor Proxy, que les permita el acceso a los equipos que no tienen salida directa. Este equipo saldrá por el primer canal de Internet, por el *gateway 192.168.1.1*.

El enrutador ubicado en la red LAN tendrá dirección IP *172.16.1.1*, y este será la puerta de enlace para poder llegar a las regionales. Los clientes únicamente tendrán como default gateway el firewall, y este a su vez tendrá la tabla de enrutamiento para poder brindar acceso a Internet, junto con la información de rutas para llegar a las regionales a través del enrutador ubicado en la LAN.

El primer paso debe ser tener el firewall Linux configurado con la siguiente información de red:

```
Nombre interface= eth0
IP=192.168.1.70/24
Nombre interface= eth1
IP=172.16.1.70/24
Nombre interface= eth2
IP=10.10.10.70/24
Nombre interface= eth3
IP=192.168.9.70/24
```

Se debe editar el archivo */etc/iproute2/rt_tables* e ingresar la información con las tablas que se utilizarán:

```
201    T1
202    T2
```

Para este ejemplo se utilizaran dos tablas. T1 y T2. Los nombres de las tablas no son importantes, se les puede dar el nombre que se desee, siempre y cuando se ingresen en el *archivo /etc/iproute2/rt_tables*, y después sean utilizadas en el script de enrutamiento. Para este ejemplo T1 será la tabla con la cual se utilizara el primer canal de Internet, cuyo *default gateway* es *192.168.1.1*. Y T2 será la tabla de enrutamiento que utilizara el segundo canal de Internet con default gateway *192.168.9.111*. Una vez se tienen estos datos configurados, se debe generar un script para la tabla de enrutamiento con la siguiente información (ver Listado 1).

Después de tener listo el enrutamiento, se deben crear las reglas de firewall para permitir el tráfico, junto con las reglas de NAT para completar el ejercicio. Para esto se crea otro script con las políticas de firewall. Este es el strip creado, donde únicamente se le asignas permisos de salida a las ips del ejercicio, junto con las reglas de NAT para este ejemplo (ver Listado 2).

NOTA: Para este ejemplo se deben tener en cuenta las siguientes condiciones:

- La `IP= 10.10.10.20` utilizara el canal de la red *192.168.1.X*, para esto se debe crear una regla de NAT que haga NAT con la dirección IP del firewall que se encuentra en esa red, en este caso la ip 192.168.1.69.
- La `IP= 10.10.10.50` utilizara el canal de la red *192.168.9.X*, para esto se debe crear una regla de NAT que haga NAT con la dirección IP del firewall que se encuentra en esa red, en este caso la ip *192.168.9.69*.
- Si se desea hacer un cambio, se debe tener en cuenta el canal a utilizar, para crear la regla de NAT respectiva con la dirección IP de ese segmento.
- Los scripts generados anteriormente se pueden crear */etc/init.d/rutas.sh* y */etc/init.d/firewall* con permisos de ejecución y para activarlos automáticamente cuando se inicia la máquina se pueden ejecutar los siguientes comandos:

```
Chkconfig --level 35 rutas on
Chkconfig --level 35 firewall on
```

Conclusión

Iproute2 permite a los administradores de redes poder tener disponibilidad de acceso a Internet, adquiriendo varios canales y utilizándoles simultáneamente, y en caso de falla de alguno de ellos poder enlutar todo el tráfico por el canal disponible con unos pocos pasos y así no perder conectividad. Este proyecto va más allá de puro enrutamiento, permite tener control sobre el tráfico TCP/IP en nuestra máquina Linux.

La gran mayoría de los sistemas actuales están limitados a tener únicamente a nivel de red una única salida a Internet (*default gateway*), por lo que con los cambios tecnológicos y de infraestructura de los últimos años esta funcionalidad se esta volviendo obsoleta, haciendo necesario que reflexionemos sobre buscar alternativas a sistemas que ya no dependan únicamente de un default gateway, si no que tengan la posibilidad de suplir esta necesidad, ya que las compañías requieren siempre estar en línea y no se permite tener excusa que por falla en los canales de comunicaciones la operación de la compañía se vea interrumpida. Siempre se debe tener presenta que una falla puede ocurrir y se debe estar preparado en como solucionarla en el menor tiempo posible y con los mínimos recursos necesarios.

Con herramientas open source se puede implementar una solución de este estilo, para lo cual se requiere de ánimo para empezar a leer y documentarse al respecto, y de tiempo para realizar pruebas. Las posibilidades que nos presenta Linux con estas herramientas son muy grandes y se pueden realizar cosas muy exitosas con muy pocos recursos.

Sobre el autor

Samuel Diaz
samuel.diaz@globalteksecurity.com
hsdiaz80@yahoo.com

Easy Peasy:
Exprime todo el potencial de tu Eee PC

Leonel Iván Saafigueroa

La distribución conocida como *Ubuntu Eee*, fruto de un proyecto independiente que pretende hacer funcionar Ubuntu (de Canonical) en los ordenadores ultra portátiles EeePC de Asus, cambia de nombre para evitar problemas con tales marcas registradas; de esta forma nace la primera versión de Easy Peasy.



linux@software.com.pl

Eee PC *Easy to Learn, Work and Play* (Fácil de aprender, trabajar y jugar), éste fue el eslogan que eligió Asus para promocionar su producto antes de que saliera de su línea de ensamblaje, algo que prometía, y mucho. Este fue el primer intento mundial por lanzar al mercado un ordenador económico, potente y que incluía como sistema operativo un GNU/Linux adaptado.

Luego otras empresas, como MSI y Dell seguirían sus pasos, aunque la Asus siempre mantendría ventaja en precios con su pequeña Eee PC 4G (la más pequeña de todas), de la cual nos ocuparemos en esta nota, veremos cómo reemplazar su sistema operativo por un Ubuntu adaptado, que reconocerá todo su hardware para que podamos usarla en todo su esplendor.

EeeC 4G

El hardware originalmente utilizado para adaptar el sistema operativo fue el Eee PC 4G, un ordenador de máxima portabilidad, con gran variedad de usos posibles, memoria fácilmente escalable, un modelo capaz de funcionar tanto con Windows XP como con GNU-LINUX, muy atractivo por fuera y que viene en distintos colores: blanco perlado, negro, rosa, verde y celeste.

Este ordenador funciona con un microprocesador Intel® Celeron M 353, el cual permite unos 900 MHz (aunque para evitar un consumo excesivo tiene una limitación de 600 MHz), su memoria de trabajo es una memoria DDR2 SO-DIMM de 512MB, y para almacenar su sistema operativo y archivos de usuario viene provisto por una memoria Flash de estado sólido (muy rápida) de 4GB, su pantalla LCD es de 7 pulgadas con una resolución nativa de 800X480 (WVGA), incluye una web cam de 0.3M Pixel, wireless interno 802.11b/g, altavoces, micrófono, tres USB para expansión, y para su alimentación incluye una batería de 4 células de 4400mAH -5200 mAH (que según el manual nos permitirá 3,5 horas de autonomía, aunque usándola realmente sin contar el modo en reposo realmente no supera las 2 horas).

Sus medidas son: 225mm x 160mm x 20mm~32mm, su peso total (con la batería instalada) es de 922 gramos; todo esto la convierte en una pequeña y potente solución para todos aquellos que busquen algo fácil de llevar a todas partes.

¿Por qué instalar Easy Peasy?

La Eee PC trae instalado de fábrica un sistema operativo GNU/Linux Xandros adaptado, el mismo convierte al pe-

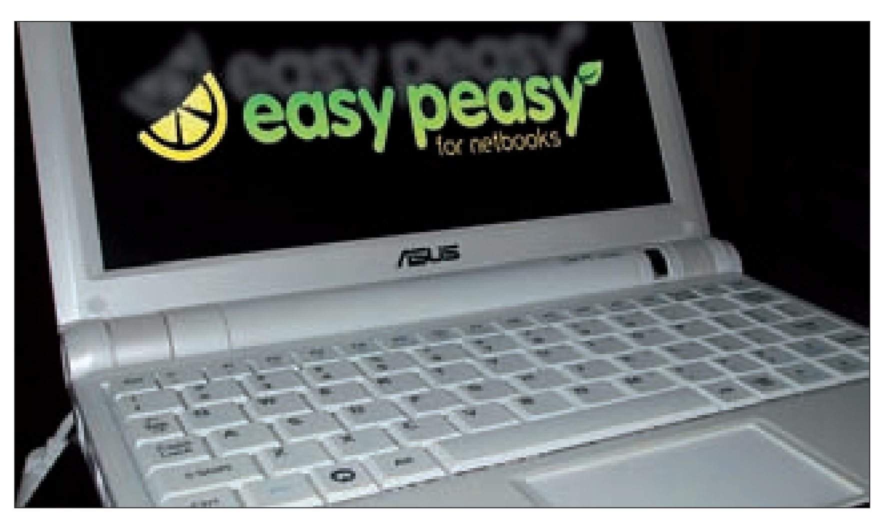


Figura 1. Easy Peasy en EeePC

queño ordenador en lo que podría ser una super agenda. Esto para un usuario novato puede estar muy bien, quien sin saber nada de informática podrá usarla de forma intuitiva, pero para alguien con más experiencia en el mundo de los sistemas operativos... seguramente se sentirá limitado, por ello para sacar mayor utilidad, Easy Peasy es la mejor opción.

Nos permitirá tener un sistema Ubuntu con todos los drivers perfectamente adaptados, como si el software fuera un guante especialmente diseñado para nuestro hardware.

Programas incluidos en Easy Peasy

Easy Peasy está basado en la última versión de Ubuntu (la 8.10), y no difiere mucho de una versión normal, lo interesante de Easy Peasy es que muchos problemas que no se puede resolver con su distribución favorita, aquí ya están resueltos.

Entre los programas que incluye podemos encontrar en accesorios: el administrador de trabajos de impresión, analizador de uso de disco, calculadoras, editor de texto, terminal, y todas aquellas cosas que traen los sistemas modernos de hoy en día. Algo curioso es el acceso universal que viene incluido con el lector de pantallas y magnificador Orca, con esto nuestro ordenador estará dotado de una voz robótica que sera útil para todas las personas no videntes, o si deseamos que nuestro ordenador nos lea en voz alta algún texto, con *espeak* también sera posible.

En el apartado de gráficos encontramos a Picasa, escáner de imágenes *xsane*, fotomatón de cámara web *Cheese* (con este programa podremos capturar vídeos y fotos desde la web cam), también en la misma sección encontramos OpenOffice dibujos.

Para Internet incluye el cliente de Bittorrent Transmission, cliente de mensajería de Internet Pidgin, cliente de Terminal Server, programa para correo electrónica Evolution, navegador web Firefox en su versión 3.0.5, Skype (versión 2.0 en español), Sun Java 6 Web Start y un visor de escritorios remotos. Para todo lo referente a trabajos de oficina trae la nueva versión de OpenOffice.org 3.0. En sonido y vídeo: Grabador de sonido, reproductor de películas Totem, y Songbird. Por último, unos cuantos juegos básicos para nuestros momentos de ocio.

Patentes y limitaciones

Esta es una distribución bastante experimental y aunque la mayoría de los programas propietarios incluidos muestran su licencia y piden conformidad antes de usarlos, no podría afirmar que tan legal es incluir Java, Flash y plugins multimedia instalados por defecto.

Fallas de la primera versión

En esta primera versión hay muchas fallas, aun podemos ver el logo de Ubuntu por todos lados, el instalador se ejecuta aun cuando el sistema ya fue instalado. Lo último se soluciona desinstalando el paquete `ubiquity`. Por lo demás todo parece funcionar bien, el hardware es detectado y no trae mayores problemas.

¿Dónde lo consigo y cómo lo instalo?

Desde su página de internet pueden descargarse de forma directa o a través de Bittorrent la imagen ISO del sistema, ésta pesa unos 900 megas aproximadamente. La forma de instalarlo es a través de una memoria SD o USB, podremos pasar la ISO a uno de estos dispositivos utili-



Figura 2. Logo del proyecto

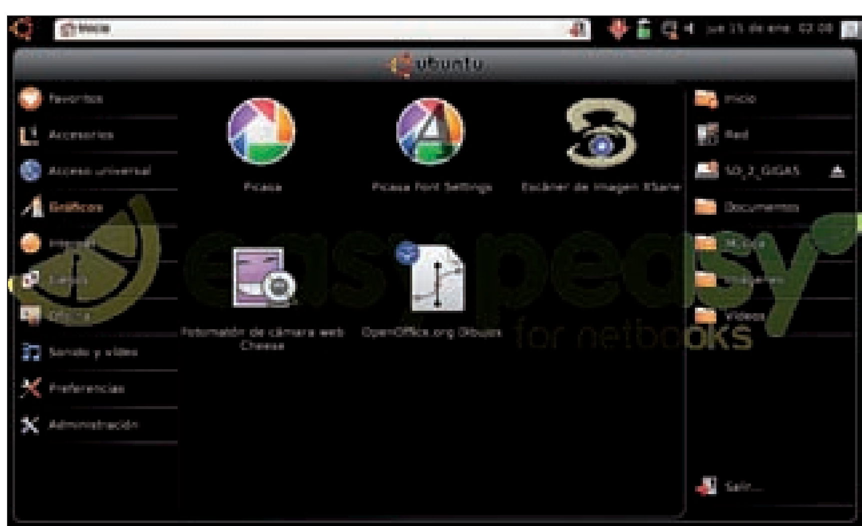


Figura 3. Entorno de usuario de Easy Peasy





Sobre el autor

Leonel Iván Saafigueroa es analista de Sistemas, docente, radioaficionado (LU5-ENP), consultor en informática y conductor del programa de radio libre hispano – Red-Handed Radio (*www.red-handed-radio.com.ar*). Si quieres hacerle algún comentario, escribe a: *leonel@saafigueroa.com.ar*

zando el programa *USB-creator*, el mismo está disponible a partir de la versión 8.10 en cualquier Ubuntu y es muy fácil de utilizar.

Luego, antes de arrancar la Eee PC, introducimos la Memoria SD/USB, luego encendemos, presionamos varias veces la tecla *ESC*, y elegimos arrancar desde el dispositivo en donde tengamos el sistema preparado; la instalación será exactamente igual a instalar Ubuntu desde CD, también podremos probar el sistema antes de decidir instalarlo.

Conclusiones

Cuando adquirí mi Eee PC, estuve toda una noche adaptando Ubuntu para que funcionara como en mi máquina de escritorio, fue divertido y aprendí mucho, creo que a cualquier usuario le encantaría pasar por una experiencia así; ahora cuando tenemos que ir actualizando el sistema operativo por uno más nuevo, es posible que no tengamos ganas de pasar por todos los problemas una y otra vez, aquí es donde proyectos como Easy Peasy, nos hace la vida mucho mas fácil.

Y éste no es el único proyecto que intenta resolver nuestros problemas, el proyecto eeebuntu (*http://www.eeebuntu.org/*), que por cierto muchas veces confundí con el que fuera originalmente Easy Peasy pero de nombre invertido, también nos provee en su web de tres imágenes distintas para poder instalar Ubuntu en nuestro ultra portátil, pero eso es algo que quizás comentaré en otra nota.

Espero que los poseedores de este pequeño ordenador puedan hacer sus propias pruebas y así puedan enviarme sus comentarios por mail.



En la red

- Página oficial del proyecto Easy Peasy: *http://www.geteasypeasy.com/*
- Página oficial de Asus EeePC: *http://eeepc.asus.com/*



Figura 4. Los distintos programas para Internet



Figura 5. Página oficial del proyecto Easy Peasy



Figura 6. Página oficial de Asus EeePC


chicas
linux
www.chicaslinux.org

Financiación de Proyectos de Software Libre

José Carlos Cortizo Pérez

El Software Libre ya se encuentra presente en el día a día de la mayoría de las personas que utilizamos ordenadores, con aplicaciones como Mozilla Firefox, Open Office o Apache.



linux@software.com.pl

Estas aplicaciones de Software Libre se ofrecen, en la mayoría de los casos, de forma gratuita, pero requieren de un gran esfuerzo para su realización, por lo que esto nos lleva a pensar: ¿cómo consiguen estos proyectos el dinero necesario para llevar adelante sus proyectos? En este artículo veremos algunas de las principales opciones que toman los grandes desarrollos de Software Libre.

Introducción

El software libre es una corriente de desarrollo de software que se fundamenta en otorgar libertad a los usuarios para que puedan usar, estudiar, copiar o modificar el software que han obtenido. Con estas libertades se promueve la concepción del software como una obra intelectual, con una cierta vida propia, no como una mera herramienta de soporte sin más, y de esta manera pequeños proyectos, como un mini kernel desarrollado en 1991 por Linus Torvalds, se han convertido en proyectos gigantescos conocidos por todo el mundo, como es el caso de GNU/Linux. Si bien el software libre permite obtener beneficio económico de la venta de un software, la gran mayoría del software libre, y particularmente los grandes proyectos de software libre, además de libres son gratuitos, lo que ha contribuido a su gran difusión.

La filosofía del software libre es apoyada desde muchos sectores como una alternativa real para el desarrollo y la evolución de la tecnología. Algunos de los proyectos de código abierto han logrado una gran popularidad pero la pregunta que algunos nos asalta es de dónde sacan la financiación estas empresas para ofrecer estos productos a los usuarios, sobre todo teniendo en cuenta que la gran mayoría de ellos se ofrecen de forma gratuita.

Formas fundamentales de financiación

Existen 5 formas fundamentales de financiación para estros proyectos:

- Donaciones de particulares o empresas interesadas en el desarrollo del software.
- Esponsorización de productos.
- Cobro por funcionalidades premium, con un enfoque muy similar al modelo freemium, muy habitual en Internet.



Este artículo es Libre

Los contenidos de este artículo son totalmente libres. Tanto el texto como las fotografías están licenciados bajo licencia Creative Commons. La editorial ha tenido a bien el comprometerse con la causa del conocimiento libre y liberar, en su Web, los contenidos de esta sección regular, dos meses después de aparecer su versión física en los kioscos.

- Oferta de servicios adicionales y labores de consultoría a terceros.
- Financiación pública.

Donaciones

A continuación están enumerados proyectos libres que se fundamentan en las donaciones.

Mozilla

Mozilla (*http://www.mozilla.org*) es uno de los casos de desarrollo de software libre sustentado por las donaciones. Los desarrollos de Mozilla Foundation, el navegador Firefox (*http://www.mozilla.com/firefox*) o el cliente de correo electrónico Thunderbird (*http://www.mozilla.com/thunderbird/*) parten de Mozilla Corporation, una entidad que nació en Agosto de 2005 y ha crecido enormemente desde entonces hasta superar los 100 empleados. Los ingresos que registra corren a cargo de donaciones privadas, siendo la más cuantiosa y famosa la que hace Google.

Cuando Google este año lanzó su propio navegador, Chrome (*http://www.google.com/chrome*), las especulaciones no se hicieron esperar. Opiniones aparte, lo que sí parece cierto es que hasta 2011 está dispuesta a seguir brindando su apoyo a Mozilla, después ya veremos. Siguiendo hablando de Google Chrome, resulta un claro ejemplo de un desarrollo realizado por una empresa que se libera para ofrecer un servicio a los usuarios que les ayuda en su línea de negocio principal. Google obtiene su principal fuente de ingresos de la publicidad en Internet, por lo que les interesa que los usuarios naveguen y se encuentren lo más cómodos posibles haciéndolo, por lo que resulta natural para su negocio el que apoyen el desarrollo de navegadores por parte de terceros (como es el caso de Mozilla) o el suyo propio (Chrome).

Ubuntu

El proyecto Ubuntu (*http://www.ubuntu.com/*), una de las distribuciones Linux más populares debido a su sencillez a la hora de instalar y utilizarse, depende de la compañía Canonical (*http://www.canonical.com/*). La base económica que mantienen en pie uno de los proyectos GNU/Linux más populares corresponden a la fortuna personal del sudafricano, Mark Shuttleworth, pionero en certificados digitales y criptografía, y primer turista espacial. Asegura seguir aportando capital personal para sustentar este proyecto en el cual tiene depositada la confianza para que un futuro logre ser autosuficiente gracias a los ingresos de empresas interesadas en servicios técnicos.

Otros proyectos

Otros proyectos financiados por aportaciones voluntarias son:

- GIMP (*http://www.gimp.org/*),
- KDE (*http://www.kde.org/*)
- GNOME (*http://www.gnome.org/*).



Figura 1. Mozilla es conocida, fundamentalmente, por su excelente navegador Firefox, aunque también desarrollan otros productos como Thunderbird. Fotografía bajo licencia CC de flod (*http://flickr.com/photos/flod/*)



Figura 2. Mark Shuttlerworth, con una camiseta de KDE bajo su camisa, en una conferencia sobre Linux. Fotografía bajo licencia CC de Stopped (*http://flickr.com/photos/foobarbaz/*)

Además estas últimas también cuentan con personas y/o compañías que esponsorizan parte del desarrollo como el mismo Mark Shuttlerworth en KDE o Google y Motorola en GNOME.

Esponsorización

A continuación están enumerados proyectos libres que se fundamentan en la esponsorización.

Apache

El servidor web Apache (*http://httpd.apache.org/*) cuenta con un organizado sistema de esponsorización; dependiendo de la cifra aportada se puede obtener la categoría Platinum, Gold o Silver; de nuevo Google se encuentra entre ellos junto con Microsoft y Yahoo.

LifeRay

LifeRay (*http://www.liferay.com*) es otro caso de compañía de Software Libre que admite la esponsorización para sustentarse económicamente. De una forma similar a la de Apache, admite 3 niveles de esponsorización por parte de empresas que, de esta manera, adquieren visibilidad y relevancia a la hora de implantar sus soluciones.



Figura 3. Logo de Red Hat

Modelo freemium

Muchas empresas dedicadas al software libre optan por un modelo que podríamos denominar freemium.

RedHat

Por ejemplo, Red Hat (*http://www.redhat.com/*), una compañía dedicada a producir una de las distribuciones de GNU/Linux más utilizadas en el entorno empresarial, ofrecen una versión gratuita para la comunidad y otra versión de pago para las empresas, en la que se aporta valor añadido con algunas funcionalidades extras y con algo de soporte. Red Hat lo hace con la mayoría de sus productos, ofertan una versión de GNU/Linux para la comunidad, que se llama Fedora (*http://fedoraproject.org/*), y otra para las empresas, que es Red Hat Enterprise (*http://www.redhat.com/rhel/*). También ofrecen su servidor de aplicaciones Java en dos versiones, JBoss (*http://www.jboss.org/*) de forma gratuita y JBoss Enterprise (*http://www.jboss.com/products/index*) para las empresas, en el que se da soporte de actualizaciones, solución de bugs, etc.

MySQL

El caso de MySQL (*http://www.mysql.com/*), un gestor de bases de datos muy popular en Internet, es similar al de Red Hat. Hace poco que fue adquirida por Sun (*http://www.sun.com/*), uno de los gigantes del software que más se ha movido hacia el Software Libre últimamente, pero desde antes de su adquisición ya presentaba dos versiones de su Base de Datos. Una versión gratuita llamada MySql Community Server y una versión de pago, para empresas, MySql Enterprise.

Oferta de otros servicios y consultoría

Muchas empresas del mundo del Software Libre siguen un modelo de ingresos basado en la consultoría, un modelo bastante lógico para este tipo de software. La idea que reside detrás de este modelo es el ofrecer el software de forma libre, e incluso gratuita, pero cobrar por instalarlo, configurarlo, adaptarlo, etc. A las empresas que implantan el software les suele dar una mayor confianza el que sean los propios creadores del software quienes se lo pongan en producción.

Sun Microsystems

Dos gigantes del software, como son Sun e IBM (*http://www.ibm.com*) siguen en gran medida este modelo de ingresos. En los últimos tiempos Sun ha dado un gran giro a su orientación, liberando casi todo el software que desarrolla, desde su lenguaje de programación Java, hasta su sistema operativo Solaris (su versión libre se llama Open Solaris, *http://www.opensolaris.org*), e incluyendo la liberación de las especificaciones de sus máquinas, con el proyecto Open Sparc.

IBM

IBM también está apostando en serio por el Software Libre, ofreciendo muchos de sus productos de forma libre, sirviendo de ejemplos su servidor de aplicaciones WebSphere (*http://www.ibm.com/websphere*), o la liberación de Lotus Notes (*http:// www.ibm.com/software/sw-lotus/*), su suite ofimática. Además de ofrecer servicios de consultoría como



Figura 4. Jonathan Schwartz (CEO de Sun) y Marten Mickos (CEO de MySQL) estrechando sus manos tras la adquisición de MySQL por parte de Sun



Figura 5. Jorge Cortell (Kanteron Systems) dando una charla en las Jornadas de Conocimiento Libre de la UEM

LISTSERV® for Linux ...
... an idea to warm up to

The power and performance critical email lists need

The original builder of email communities with its invention in 1986, L-Soft's LISTSERV is today the leading industrial-strength email list manager. LISTSERV offers state-of-the-art deliverability features and is the only email list software with the security of integrated virus protection. LISTSERV can be controlled from anywhere on the Internet through its fully customizable Web interface. LISTSERV is renowned for its flexibility, scalability and performance.

More than 3,700 organizations and businesses worldwide trust LISTSERV for their critical email list needs. ***Isn't it time you turned up the heat on your email challenges?***

LISTSERV for Linux is available for i386, 64-bit and S/390 architectures.
Virus protection is provided by the integrated F-Secure® Anti-Virus.
LISTSERV Free Edition is available for non-profit hobby users.
LISTSERV is available only from L-Soft.







Figura 6. Alvaro Lopez (Cherokee, Octality), durante una presentación sobre Cherokee en las Jornadas de Conocimiento Libre de la UEM

fuente principal de ingresos, en la mayoría de los casos también ofrecen versiones *premium* de su software.

Empresas españolas

En España ya existen varias empresas que viven de los servicios de consultoría sobre Software Libre, como es el caso de:

- Kanteron Systems (*http://www.kanteron.com*) que ofrece software para estaciones de imágenes médicas,
- Octality (*http://www.octality.com*) que se dedica a ofrecer soluciones de alto rendimiento y ha surgido como empresa de consultoría asociada al proyecto Cherokee (desarrollado por Álvaro López el CTO de Octality),
- Stratebi (*http://www.stratebi.com*), una empresa dedicada a ofrecer consultoría de Pentaho (*http://www.pentaho.com*), una suite Open Source de Business Intelligence.

En estos casos a las empresas les interesa seguir desarrollando su producto de forma libre para que obtenga mayor difusión y sea más utilizado y, de esta forma, ser ellos a los primeros que se les llame a la hora de implantar su producto u ofrecer soluciones fundamentadas en él.

Financiación Pública

Una opción que toman algunos de los proyectos de Software Libre es la financiación pública. Por ejemplo, en EEUU una de las fuentes principales de financiación de proyectos de I+D proviene del departamento de defensa o la NSF (National Science Foundation), y en muchos de los casos, los resultados de los proyectos de I+D acaban llegando al gran público al abrirse las tecnologías o en modo de Software Libre. Si a día de hoy tenemos Internet es gracias a la financiación del departamento de defensa que dio lugar a tecnologías como el TCP/IP que sustentan las conexiones que utilizamos habitualmente.

En España proyectos como el proyecto Morfeo (*http://morfeo-project.org/*) apoyan el desarrollo del Software Libre financiados por entidades como el Ministerio de Industria. Y es que, cada día más, los organismos públicos apoyan el desarrollo del Software Libre (aunque sería necesario un mayor apoyo público), en gran medida por las directrices de Europa que ven en el Software Libre una excelente forma de mantenerse al margen de los monopolios que tradicionalmente han existido en la industria del software, así como una forma de desarrollar estándares que permitan la interoperabilidad en áreas de mucho interés, como el e-Gobierno.

Con fondos europeos se han desarrollado proyectos de pequeña o mediana envergadura dentro del Software Libre, como el filtro de contenidos POESIA (*http://www.poesia-filter.org/*), aunque todavía falta que los proyectos de Software Libre financiados con fondos públicos logren un nivel de impacto mayor en la Sociedad.

Otras fuentes de ingresos

Además, surgen empresas o servicios que se financian a partir de modelos mixtos, combinando algunos de los modelos ya vistos, o añadiendo alguna otra forma de ingresos. Por ejemplo, eMule (*http://www.emule-project.net*) y Menéame (*http://meneame.net/*), dos proyectos que obtienen la ayuda económica a través de las donaciones y la publicidad repectivamente que, además, tienen una tienda virtual desde la que comprar objetos con el logo de le empresa, otra forma de lograr financiación y publicidad.



Figura 7. Emilio Arias (Stratebi) haciendo una demo de las soluciones que ofrece Stratebi en las Jornadas de Conocimiento Libre de la UEM

Sobre el autor

José Carlos Cortizo Pérez es Ingeniero Superior en Informática y, actualmente, doctorando en el programa de Ciencias de la Computación de la Universidad Carlos III de Madrid. Tras una dilatada experiencia como consultor, actualmente compagina sus labores como CTO en AINetSolutions y Wipley, con labores docentes y de investigación en la Universidad Europea de Madrid. Su bitácora personal se encuentra en *http://josekblog.blogspot.com*

Por otra parte, José Carlos es miembro de GLUEM (Grupo de Usuarios de Linux y Software Libre de la Universidad Europea de Madrid, *http://www.gluem.org*), organización muy activa que, de forma anual, organiza las Jornadas de Informática de la UEM (en Otoño) y las Jornadas de Conocimiento Libre (en Primavera).



Figura 9. Financiación de proyectos de Software Libre, blog de Emilio Marquez

Conclusiones

De todos estos datos se pueden sacar dos conclusiones fundamentales:

- Los proyectos de Software Libre son económicamente viables y permiten a las empresas vivir más allá de la venta del propio software
- Los grandes de la informática, incluyendo los grandes de Internet, tienen mucho que ver en estos proyectos.

Google, Yahoo, IBM, Sun, etc. están detrás de muchos de los proyectos, ya que está claro que del aire pocas cosas viven. ¿Y por qué los grandes apoyan estos proyectos? Por muchos motivos, entre los que encontramos que en muchos casos el software que liberan es un software que puede ayudar a mejorar los ingresos de su área de negocio principal (caso de Google Chrome), que los servicios premium son los que realmente sustentan su economía, o incluso como una forma de competir con otros productos más asentados (como es el caso de IBM con su Lotus Notes compitiendo con el Office de Microsoft).

Por otra parte, conviene ser conscientes en cuanto a que los proyectos de Software Libre, a pesar del apoyo de la comunidad de desarrolladores Open Source, necesitan de financiación económica para muchas tareas, desde la difusión de sus productos, posting para albergar el software, defensa de sus productos ante terceros que quieran violar las licencias del software, o incluso el desarrollo de tareas concretas y/o críticas que no están siendo abordadas por la comunidad.

El conocer las distintas alternativas de financiación de los proyectos actuales de Software Libre también nos permite tener una visión más clara de las fuentes de financiación aplicables a los proyectos en los que colaboramos porque nunca se sabe cuándo podremos necesitar financiación adicional para conseguir que un proyecto llegue más lejos de lo que jamás habíamos pensado.

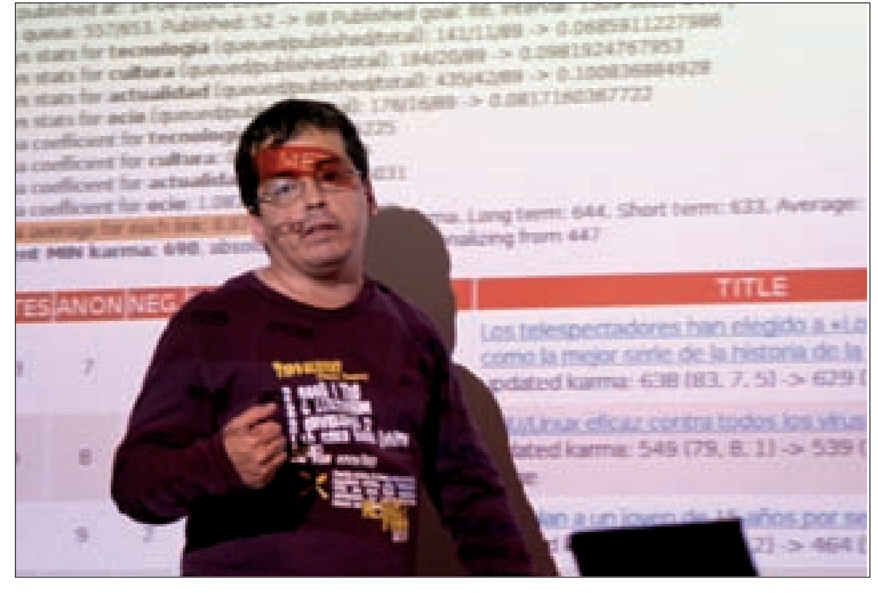


Figura 8. Ricardo Galli (Meneame) presentando Meneame.net en las Jornadas de Conocimiento Libre de la UEM



En la red

- Financiación de proyectos de Software Libre, blog de Emilio Marquez: *http://tinyurl.com/9gpt3e*
- La financiación de los proyectos de Software Libre, BULMA: *http://tinyurl.com/9rcomz*
- Así se financian los grandes proyectos del Software Libre, Abadía Digital: *http://tinyurl.com/7xg74e*
- La financiación en los proyectos Open Source, SoftwareLibre.net: *http://tinyurl.com/83utrf*

Primeros pasos
con Scribus

Diego Germán Gonzalez

Si hay una herramienta infaltable en el disco rígido de quienes usamos el ordenador para trabajar o estudiar, es un buen software de autoedición (sae). Existen muy pocas actividades en las que no se utilice material impreso ya sea para presentar informes, hacer anuncios, comunicar ideas o pedir datos.



linux@software.com.pl

Si tenemos en cuenta que los destinatarios de nuestros esfuerzos de comunicación impresa tienen mucho más material que leer, una buena diagramación que facilite la lectura no solo es un acto de cortesía, sino una necesidad para lograr nuestros objetivos. Por suerte, los usuarios de Linux contamos con una muy eficaz herramienta de autoedición incluida en los repositorios de la mayoría de las distribuciones importantes: Scribus.

Introducción

¿Por qué molestarse en instalar y aprender a utilizar un programa para hacer cosas que se pueden hacer muy bien con OpenOffice? Muy sencillo, para hacer cosas que no se pueden hacer muy bien con OpenOffice. Un procesador de textos se ocupa del contenido, un programa de autoedición: de la disposición de ese contenido. OpenOffice resulta adecuado si nuestra diagramación consiste en disponer el texto de izquierda a derecha y de arriba para abajo, pero si vamos a adoptar disposiciones diferentes o incluir varias imágenes la cosa se complica.

Descubrí la importancia de una buena diagramación en la adolescencia. Cuando yo estaba en el colegio secundario podía elegir entre dos libros de matemáticas, el de Repetto y el de Tapia. El primero estaba impreso a un solo color, con una tipografía diminuta, el segundo tenía letras grandes y diferenciaba con colores distintos el planteo, el desarrollo y las soluciones. En principio la forma de impresión no debería importar, la demostración del teorema de Pitágoras era la misma. Sin embargo, con Tapia yo obtenía mejores notas.

Cursando mis estudios universitarios descubrí que tardaba menos tiempo estudiando con libros importados que con los de autores nacionales. La diferencia estaba por la distribución del texto, en los extranjeros las páginas eran más anchas lo que permitía destacar en el margen los puntos importantes, poner un recuadro ejemplificando un punto importante y utilizar distinta tipografía. Los nacionales eran páginas y páginas de bloques de texto interrumpidos por un gráfico que casi nunca estaba cerca del texto correspondiente.

En la actualidad concurro a oficinas públicas donde me hacen llenar formularios con campos de texto diminutos en comparación con la información pedida, recibo facturas donde el único dato legible es el importe a pagar o navego por páginas web donde la información que necesito está en el

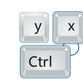




Figura 1. Configuración de Scribus

lugar más inesperado. Este artículo está dirigido a los usuarios comunes que quieren mejorar el resultado de sus impresos, estoy muy lejos de saber algo de diseño gráfico por lo que remito a los profesionales del tema a la sección *En la red* para encontrar la información que necesiten.

Instalando Scribus

Como dijimos al principio, Scribus se encuentra en los repositorios de las más importantes distribuciones Linux, la última versión disponible en la cual se desarrollan los ejemplos de este artículo es la 1-3-3.12. Además, hay versiones para Windows y Mac.

En la web del desarrollador recomiendan la instalación de los siguientes paquetes:

- SCRIBUS: La última versión estable, al momento de escribir este artículo la 1.3.3.12,
- SCRIBUS-TEMPLATES: Modelos de páginas adicionales,
- SCRIBUS-DOC: La documentación del programa,
- ICC-PROFILES: Colección de perfiles de color para utilizar con Scribus y otros programas,
- IPROF: Creador de perfiles de colores para utilizar con nuestro hardware,
- QT3-QTCONFIG: Solo usuarios de gnome para poder visualizar correctamente la interfaz gráfica del programa,
- GHOSTSCRIPT: Intérprete postscript.

Software sugerido:

- Tipografias:
  - t1-xfree86-nonfree,
  - ttf-bitstream-vera,
  - ttf-gentium,
  - ttf-xfree86-nonfree.
- Otras herramientas:
  - Visor pdf (acrobat reader, xpdf, etc),
  - Gimp,
  - IMAGEMAGIK: conversor de formato de imágenes,
  - INKSCAPE: programa de diseño de gráficos vectoriales.

Anuncio de perro perdido

Una vez instalado e iniciado el programa, lo primero que nos aparece es la pantalla de configuración del documento, Está en castellano y es bastante comprensible por lo que sólo nos limitaremos a hacer una recomendación. Si vamos a hacer un documento destinado a ser impreso, utilicemos centímetros como unidad de medida. Si utilizamos una versión electrónica: el punto.

El diseño de una página en Scribus se basa en marcos; estos pueden colocarse en cualquier sector de la página, son de tamaño y color configurables y pueden contener texto e imágenes.

Nuestro primer proyecto será crear un cartel pidiendo ayuda para la búsqueda de un perro perdido. Solo estará compuesto por tres elementos: la foto, el texto y el borde del cartel.

Comenzamos creando el marco contenedor de la imágen. Para esto sólo debemos presionar la tecla “*I*” y luego marcar un sector del área de trabajo con el puntero mientras mantenemos presionado el botón izquierdo del ratón. Soltamos y cargamos la imágen con *CRTL+D*.

Muy dificilmente el tamaño del marco coincida con el de la fotografia, eso se soluciona yendo al menú *Objeto->Ajustar->Marco a imágen*. Si lo que queremos es centrar la imagen, podemos hacerlo con la ayuda de las guías: *Menú->Vista->Mostrar guias->Mostrar cuadriculas*. Una vez colocada en la posición deseada, bloqueamos la imágen haciendo click en el botón derecho y seleccionando *Bloqueado*.

Nuestro segundo paso es crear el contenedor para el texto con la información sobre el animal extraviado. Con el puntero sobre el área de trabajo presionamos la letra “*T*” y seleccionamos un espacio del ancho del campo y de un largo suficiente para contener lo que vamos a escribir. De todas formas, siempre podremos modificar el tamaño del marco.

Por tratarse de frases cortas utilizaremos el editor de texto de Scribus. Éste a primera vista dispone de las mismas herramientas de un pequeño procesador de texto, pero con la desventaja de que uno no puede visualizar en su ventana las modificaciones de estilo que hace. Aclaremos esto: si nosotros en OpenOffice.org seleccionamos la tipografía Times New Roman en tamaño 12 y color rosa, al escribir en pantalla veremos el texto con esas características, en la ventana del editor de texto de Scribus no. Tendremos que darle al botón de *Actualizar marco de texto* para ver el resultado en el área de trabajo.

La redacción del texto consta de pocas frases. Para cada una de ellas seleccionamos una tipografía, un color y un tamaño de fuente. Presionando *CRTL+U* podemos ir viendo cómo queda en el área de trabajo. Una vez finalizado, bloqueamos el marco como hicimos con el de imágen.

El paso final es crear el borde del cartel. Para esto presionamos la letra “*S*” y movemos el puntero para abarcar toda el área de trabajo. Luego con el botón derecho presionado seleccionamos *Propiedades->Colores->Editar propiedades del color de relleno->Ninguno*.

Si queremos un borde más ancho, hacemos click en *Línea* y aumentamos el ancho.

Texto con distribución irregular

Dijimos anteriormente que una de las características de Scribus es la posibilidad de distribuir el texto en diversos lugares del área de trabajo, también comentamos que para trabajos extensos conviene utilizar un procesador de texto externo. Vamos a probar las dos cosas, para lo que necesitamos un artículo lo suficientemente largo para ocupar una página.

Ya aprendimos cómo crear un marco de texto y uno de imagen, Con el primero vamos a crear el título centrado en la parte superior y con el segundo uno de imagen, ubicado inmediatamente abajo del título y a la derecha del área de trabajo.

A continuación podemos crear un borde para el título como hicimos en el ejercicio anterior (con la diferencia que la forma sólo debe abarcar el área del título) o si queremos un fondo sólido cambiamos el texto del título a un color claro, creamos la forma y con el botón derecho vamos a *Nivel->Bajar*. Recordemos siempre que una vez que tenemos los marcos en



Figura 2. Anuncio de perro perdido terminado

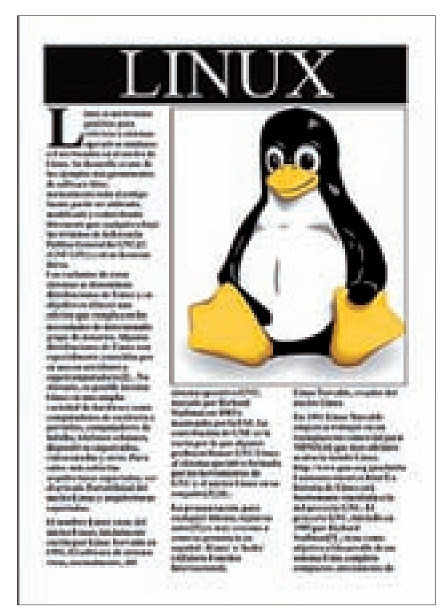


Figura 3. Text en tres columnas

la posición deseada es conveniente bloquearlo para evitar moverlos accidentalmente.

A continuación vamos a crear un marco de texto abajo del título y a la izquierda de la imagen que tenga el mismo largo del área de trabajo, no olvidemos verificar que imagen y texto deben comenzar a la misma altura en *Propiedades->X,Y,Z->Posición Y*.

El segundo marco de texto va inmediatamente debajo de la imágen por lo que debemos verificar la coincidencia de la *Posición Y*, este contenedor en particular va a tener dos columnas: *Propiedades->Forma->Columna 2*.

Para que Scribus utilice dos marcos diferentes para un mismo texto hay que enlazarlos. Esto se hace cliqueando en el primer marco, presionando la letra "*N*" y cliqueando en el segundo. Con "*C*" y el puntero indicamos el marco de inicio y con *CRTL+D* cargamos el archivo de OpenOffice. Es posible que el texto no quepa entero en el espacio asignado por lo que debemos trabajar con el tamaño de la tipografía.

Es evidente que utilizar el método de prueba y error cada vez que queremos diagramar un documento resulta muy poco práctico por lo que una vez que le tomemos la mano al programa podemos establecer estilos de párrafo con las combinaciónes de tipografía, color y tamaño que consideremos adecuada. Esto se hace desde el menú *Editar->Estilo de párrafo->Nuevo*.

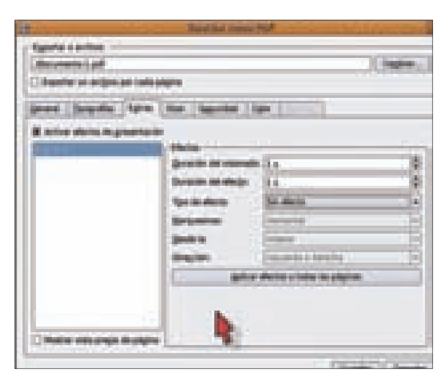


Figura 4. Presentación

Una vez establecidos los nuevos estilos sólo tenemos que aplicarlos, basta con seleccionar en el editor de textos la porción en la que queremos hacerlo, hacer click en la indicación del estilo de texto que aparece en la ventana de la izquierda y seleccionar el nuevo.

Trabajando con PDF

Uno de los problemas que tenemos los usuarios de software libre es que las herramientas de creación de pdf los hacen mucho más pesados que por ejemplo el software de Adobe. Scribus a pesar de su facilidad para exportar en ese formato tampoco es la excepción. Por suerte existe una herramienta que permite adelgazarlo: PDFtops.

El método (utilizando Ubuntu 8.10) es el siguiente:

- Instalamos el paquete `xpdf-utils`,
- Abrimos Scribus y creamos una ventana de texto que ocupe todo el ancho del área de trabajo. Con el botón derecho seleccionamos *Texto de ejemplo standar lorem ipsum*,
- Vamos a menú *Archivo->Exportar->Guardar como PDF* e ignorando el mensaje de error (únicamente en este caso porque no vamos a imprimir) lo guardamos,
- Abrimos la terminal y escribimos `pdftops -level3 documento de origen.pdf documento de destino.ps`,
- Abrimos el archivo resultante con el visor de documentos *evince* y seleccionamos *Archivo->Imprimir->Imprimir a un archivo->Formato de salida PDF*.

En nuestro caso logramos bajar el archivo de un peso de 250 kb a 32 kb.

Scribus nos permite crear archivos pdf interactivos. Comencemos con lo más fácil, es decir la creación de un link:

- Creamos un cuadro de texto en el que escribimos `Sitio web revista LiNUX +`,
- Cliqueamos en el botón *Crear anotaciones PDF* y seleccionamos *Enlace*,
- Dibujamos un recuadro alrededor del texto,
- Con el botón derecho vamos a *Opciones PDF->Propiedades de anotación->Tipo->Enlace->Web externo* y escribimos: `http://www.lpmagazine.org/prt/view/pag-prin.html`,
- Exportamos como pdf y probamos.

Existe un método más corto que es crear un cuadro de texto con el botón *Insertar campos PDF*, pero tiene la desventaja que el documento resultante solo puede verse correctamente con el Acrobat Reader.

Creando un formulario PDF

Scribus tiene una muy buena integración con javascript, lo que permite crear pdfs interactivos, esa es la buena noticia. La mala es que para verlos necesitamos el Acrobat Reader.

Vamos a crear un sencillo formulario. Creamos un marco de texto que diga `Nombre:` y a continuación presionando en el botón *Insertar campo PDF* seleccionamos *Campo de texto->Opciones PDF->Propiedades de campo*. Una vez hecho esto podremos seleccionar las opciones de estilo y las acciones a realizar una vez completado el formulario. Si conocemos javascript, las posibilidades se amplian al límite.

Presentaciones

La creación de las presentaciones pdf no traen ningún tipo de complicación. Seleccionamos una página por cada diapositiva y diagramamos utilizando los marcos en la forma que vimos anteriormente. Una vez finalizado cliqueamos en *Archivo->Exportar->Guardar como pdf->Extras->Activar efectos de presentación* y aplicamos los diferentes efectos disponibles a cada página.

Conclusiones

En este artículo sólo dimos un pequeño repaso de las posibilidades de Scribus. Invitamos a los lectores a visitar la página web de los desarrolladores para conocerlas todas. Si a eso le sumamos la posibilidad de escribir scripts en Python para incorporar nuevas funcionalidades y la interacción con GIMP, OpenOffice e Inkscape y el precio accesible de la cada vez mayor cantidad de impresoras láser que trabajan con Linux, nuestros documentos no tendrán nada que envidiarles a los salidos de las mejores imprentas industriales.



En la red

- *www.scribus.net*
- *www.http://wiki.scribus.net/index.php/Main_Page*


TODAVÍA ESTÁS
A TIEMPO
DE SALIRTE
DE LA NORMA
HACK
CRACK
VIRUS
CRIPTOGRAFÍA
RETROINFORMÁTICA
SOFTWARE LIBRE
TECNOLOGÍA
HACKTIVISMO
PROGRAMACIÓN
ALG@RROBA
@RROBA
MÁS DE 10 AÑOS HABLANDO COMO NADIE
DE SEGURIDAD INFORMÁTICA... Y DE OTRAS COSAS

Libera tus oídos

Leonel Iván Saafigueroa

Música libre y legal en la red, una tendencia que está muy de moda. El nacimiento de distintas licencias que eliminan las restricciones de distribución o modificación impuestas por el copyright en diversidad de obras, hizo posible el surgimiento de nuevos artistas y la liberación de trabajos por parte de muchos ya conocidos, todo esto gracias a la condición de que el trabajo derivado se mantenga con el mismo régimen de derechos de autor que el original.



linux@software.com.pl

La música libre es aquella música que está en dominio público o protegida por una licencia libre. De esta forma se consigue la máxima distribución y promoción de la música y se hace posible que llegue sin obstáculos al público en general.

En este artículo conocerán a las distintas licencias que hacen posible la liberación legal de contenidos libres, también servirá de guía para saber en dónde buscar y cómo publicar tu propia obra.

Fundación Copyleft

La Fundación Copyleft nace para potenciar y defender las producciones de arte, cultura y ciencia bajo licencias copyleft.

Copyleft o lo que será en español *Copia Permitida*, nace con la intención de crear distintos derechos de autor capaces de eliminar las restricciones de distribución o modificación impuestas por el copyright, con esta licencia se pretende proteger una gran variedad de obras como programas informáticos, arte, cultura y ciencia.

En su sitio web explica de esta forma que cualquier tipo de producción creativa quedaría amparada por esta licencia, dando a los autores la posibilidad de un mayor control sobre sus obras, investigaciones y proyectos, permitiendo una remuneración compensatoria más razonable por su trabajo, también permiten a los usuarios finales un mejor acceso y disfrute de los bienes bajo este tipo de licencias no restrictivas.

Copyleft es una fundación que invita a participar con el objetivo de potenciar, afianzar, organizar, incentivar, dirigir, realizar, proteger y defender la producción de cultura Copyleft, protegiendo los intereses de creadores, intérpretes, artistas, productores, y usuarios interesados en la misma.

Licencia Creative Commons

Al igual que la fundación Copyleft, en la licencia Creative Commons nos encontramos con una organización no gubernamental, sin ánimo de lucro que desarrolla planes para ayudar a reducir las barreras legales de la creatividad, por medio de una nueva legislación y nuevas tecnologías. La misma fue creada por Lawrence Lessig, un profesor de derecho de la Universidad de Stanford y especialista en ciberderecho. Una traducción de Creative Commons en español podría ser *bienes comunes creativos*.

Aunque inspirada en la licencia GPL (General Public License) de la Free Software Foundation, no es su objetivo el licenciamiento de software; esto es lo que la diferencia muy



claramente del Copyleft, siendo una licencia exclusiva de contenidos.

Existen una serie de licencias Creative Commons, cada una con diferentes configuraciones o principios, permitiendo al autor original dar libertad para citar su obra, reproducirla, crear obras derivadas, ofrecerla públicamente y con diferentes restricciones, como no permitir el uso comercial o respetar la autoría original.

La licencia Creative Commons, es una de las primeras en adaptarse a la legislación vigente de cada país, ofreciendo un verdadero respaldo legal.

¿Dónde encontrar obras libres?

Internet está repleto de lugares en donde se encuentra material libre, aquí mencionaremos los más populares, para que la búsqueda sea un poco más simple.

La mundial

Una de las bandas libres pioneras en España. En la página de estos geniales músicos españoles, nos encontramos en la sección de descargas un simpático texto que aclara: *Aquí puedes descargarte todos los temas que hemos grabado hasta el momento, están enteritos, 128kbps, 44kHz y stereo. Solamente registramos los temas para que quede claro que son nuestros pero son de libre copia, circulación y modificación mientras se cite la fuente*, esta genial banda no solo liberó su música y se ampara bajo ambas de las dos licencias mencionadas en este artículo, sino que además invita a otros a hacer lo mismo, en su propio servidor podemos encontrar alojados los trabajos de: s3r raRØ, Caladitos, BEUC Balance, Xera, La Comodidad del anillo, Error Fatal, Galligan, Gnugpl Dobroide, Mississippi Alligators, El Sr. Silicona, Tomás Pinel, Black Orchid, Canteca de Macao, David Doldán, Musicalibre.info, Debaser, Víctor Olaya, Hoka Hey, Knock Out, Explicit Language, Adore, Gaia Stalla, Rayo y los Trueno; se encuentran disponibles en la sección: BY THE FACE (free música), de su pagina web.

Los oídos más agudos escuchan JAMENDO

En todo lo que se refiere al mercado internacional de la música libre, nos encontramos con JAMENDO: *Los artistas de Jamendo te permiten escuchar, descargar y compartir su música. Es libre, legal e ilimitado*; así lo anuncia su sitio web en donde encontraremos música de los más variados estilos: metal, rock, reggae, hip hop, electro, amparándose bajo la licencia Creative Common. Jamendo nos permite llegar con solo hacer un click a distintas obras como ser las del genial concertista Rob Costlow (con geniales composiciones como *Meant to Be*, utilizada por Condor Films para el documental de Roger Federer), o artistas como Ray Grant con un estilo muy a lo Depeche Mode.

Este sitio incluye un excelente sistema que nos ayudará a encontrar lo que buscamos, mostrando un ranking de descarga e incluso podemos hacer una búsqueda utilizando las influencias de las bandas, en donde por ejemplo introduciendo *Nirvana* o *The Ramones*, nos dará como resultados distintas bandas no tan conocidas que pretenden sonar como aquellas que si lo son.

El sitio permite escuchar la música de forma online como también descargarla todo de forma gratuita y con una excelente calidad.

MySpace: Fotos y música, pero no con tanta calidad

Este es otro sitio muy de moda por estos días, en donde muchas bandas deciden darse a conocer, aquí se encontrarán todas aquellas que hagan un verdadero negocio de la música; la calidad del sonido es menor, pues la intención es la promocional de nuevos trabajos y publican aquí a modo de demo.



Figura 1. Página inicial de MySpace.com

Sobre el autor

Es analista de Sistemas, docente, radioaficionado (LU5ENP), consultor en informática y conductor del programa de radio libre hispano – Red-Handed Radio (*www.red-handed.com.ar*). Si quieres hacerle algún comentario, escribe a: *Leonel@saafigueroa.com.ar*

¿Dónde subir mi obra?

En cualquiera de los sitios mencionados anteriormente pueden registrarse y subir sus trabajos, pero no debemos olvidarnos del sitio *archive.org*, el mismo permite subir cualquier obra que sea libre, esto incluye libros, y distintos tipos de publicaciones, siendo una verdadera opción para publicar programas de radio online (podcasts), sin ninguna limitación de espacio.

Este es el sitio que yo elegí para subir desde hace ya varios años mi programa de radio online Red-Handed, y jamás tuve un problema, los programas siguen ahí, en este programa hemos hecho entrevistas hace algún tiempo atrás a la *mundial.net*, si buscan puede que lo encuentren.

Conclusiones

Actualmente estemos muy dominados por todo lo comercial, tanto en las radios como en la TV, no creo que esté mal, pero hay otro mundo ahí afuera, un mundo de geniales artistas que se merecen ser escuchados.

Espero que puedan darse una oportunidad y dar un paseo por los sitios web mencionados, y si eres un artista, espero quieras compartir y darte a conocer, un mundo lleno de posibilidades te espera, solo necesitas un navegador de Internet, ganas y ajustar tus sentidos, es gratis y además puedes elegir todo lo que quieras. 

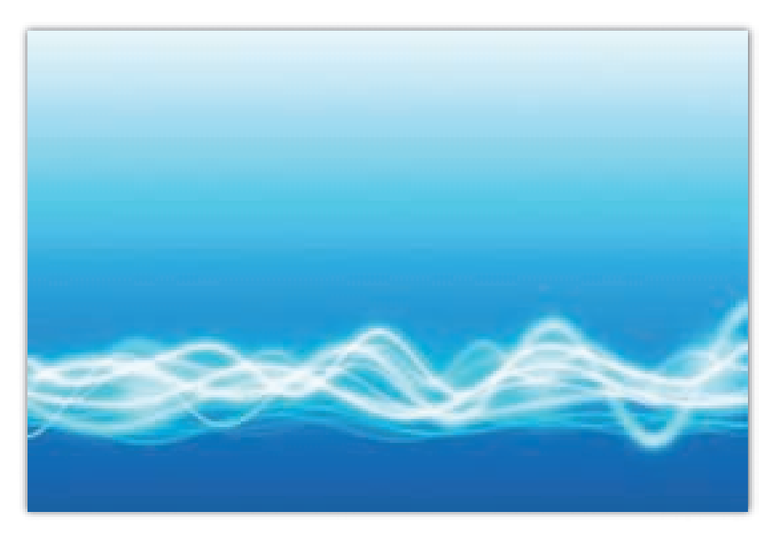
En la red

- Página oficial de la fundación copyleft: *http://www.fundacioncopyleft.org/*
- Página oficial de Creative Commonds: *http://creativecommons.org*
- Jamendo *Abre tus oídos* (Catálogo de música libre): *http://www.jamendo.com/es/*
- Sitio de música libre La Mundial: *http://lamundial.net/*
- Fotos y música de todas las bandas: *http://www.myspace.com/*
- Página de almacenamiento libre archive.org: *http://www.archive.org/*
- Programa de radio libre Hispano: *http://www.red-handed-radio.com.ar*

Estudiando con Linux

Diego Germán González

Si algo caracteriza la época que vivimos es la necesidad permanente de seguir aprendiendo. Mientras que con algunos años de escolarización nuestros abuelos eran capaces de conseguir y conservar un trabajo para toda la vida, nosotros para conservar un lugar expectante en el mercado laboral nos vemos obligados a continuar estudiando aún después de terminada la universidad.



linux@software.com.pl

El objetivo de este artículo es analizar algunas de las aplicaciones que funcionando bajo Linux nos facilitan el proceso de aprendizaje. No todos incorporamos conocimientos de la misma manera. El borrador del texto que están leyendo se escribió de memoria en una pda sin conexión a Internet ni posibilidad de consultar apuntes, al revisarlo teniendo a mi alcance las dos cosas compruebo que cometí relativamente pocos errores y omisiones. Sucede que yo recuerdo más fácilmente las cosas que escribo y una versión de este artículo fue borrada accidentalmente sin conservar copias de respaldo. Otras personas son capaces de recordar piezas musicales complejas habiéndolas escuchado una vez o identificar un ingrediente entre miles con solo probarlo, el secreto es descubrir con cuál sentido memorizamos mejor las cosas. En otras palabras no existe una receta infalible para el aprendizaje. Si tuviéramos que resumir nuestra propuesta en una frase, sería *Conócete a ti mismo y busca en los repositorios*.

Trabajando con material impreso

La mayor parte del material de estudio se conserva en papel y las diferentes distros disponen de variadas aplicaciones que nos permiten transformarlo en formato digital. Gracias al proyecto Sane, Linux soporta sin problemas la mayoría de los escáneres y existen excelentes aplicaciones para manipular las imágenes obtenidas, pero a la hora de hacer un reconocimiento óptico de textos (procedimiento muy necesario para poder editar el material de estudio o utilizarlo en alguna otra aplicación) debemos pasar mucho tiempo hasta lograr la configuración correcta.

Herramienta de escaneado y retoque

Una excelente aplicación para la digitalización es XSANE, algunas distros como Ubuntu y OpenSuse la traen instalada por defecto y figura en los repositorios de casi todas las demás. El programa puede utilizarse solo o como un plugin del Gimp lo que aumenta la cantidad de herramientas para trabajar con la imagen.

Lo primero que debemos hacer es indicarle a XSANE cómo trabajar con la imagen escaneada: las opciones son:

- *Visor interno*: muestra en pantalla la imagen escaneada para permitirnos trabajar con ella.
- *Guarda imagen*: guarda el resultado del escaneo en la dirección especificada.

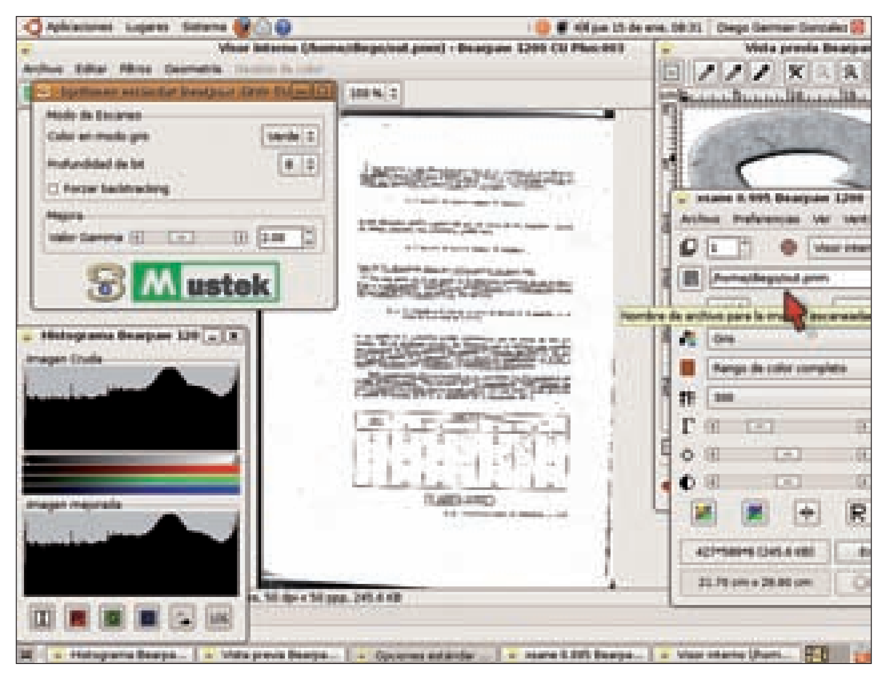


Figura 1. Xsane

- *Copia a impresora*: actúa como una máquina fotocopiadora imprimiendo directamente el documento escaneado.
- *Multipágina*: permite escanear libros y documentos de varias páginas.
- *Fax*: envía el texto escaneado por un fax módem.
- *Correo electrónico*: envía el texto escaneado vía e-mail.

No siempre la calidad del impreso a escanear es la mejor, sobre todo en países como Argentina donde la principal herramienta pedagógica es la fotocopia, la tinta se exprime hasta el límite de sus posibilidades y para bajar costos se reduce el tamaño del original hasta el límite de lo ilegible. XSANE permite corregir esto:

- *Gamma correction*: regula la intensidad de los grises.
- *Brightness*: aumenta o disminuye el brillo de la imagen.
- *Contrast*: regula el contraste de la imagen.
- *Threshold*: establece el porcentaje de brillo necesario para obtener un punto blanco.
- *RGB default*: si está activada establece los valores por defecto para los componentes en color (rojo, verde y azul) para los parámetros anteriores, si desactivamos la opción podemos fijarlo nosotros.
- *Autoenhancement*: el programa ajusta automáticamente gamma, brillo y contraste de acuerdo al documento a escanear.

La mejor herramienta para trabajar en la etapa del post escaneo hay que usarla desde la terminal, se llama *Unpaper*.

Unpaper es un soft para postprocesamiento de documentos de papel escaneados que resulta muy útil en el caso de fotocopias de libros que incluyen dos páginas del mismo por carilla, preparación de archivos gráficos para reconocimiento óptico de caracteres y eliminación de los bordes negros que suelen aparecer cuando el objeto escaneado es más chico que el área de trabajo del periférico o tuvo que variarse de posición para que entrara en ésta, sobre todo para páginas de libros que han sido exploradas de fotocopias antes creadas. Para poder utilizar *Unpaper* debemos guardar los archivos escaneados en alguno de estos formatos: pbm, pgm o ppm.

Abrimos una terminal y escribimos: unpaper opción archivo a transformar archivo transformado.

Si los archivos no están en el directorio personal y no cambiamos de directorio con el comando `cd` debemos poner la ruta completa, por ejemplo: `/home/micarpeta/Desktop/prueba.ppm`

Algunas opciones disponibles son:

- *Layout*: cantidad de páginas a mostrar por hoja, si van a estar rotadas con respecto al original y en qué sentido con respecto a las agujas del reloj.
- *Start*: en el caso de que el archivo de inicio sea multipágina la página de inicio del escaneo.
- *End*: lo mismo pero determina la página del final.
- En ambos casos si en cada hoja hay dos carillas podemos determinar una de las dos como inicio o final:
- *Exclude*: excluye una o varias páginas del procesamiento del documento.
- *Size*: determina el tamaño del documento resultante.

En el segundo párrafo de la nota sostuvimos que cada uno de nosotros tiene una manera diferente de incorporar conocimientos, si por ejemplo aprendemos mejor escuchando un audio en nuestro reproductor personal mientras trotamos por el parque o tenemos una fuerte memoria visual, necesitamos cambiar los contenidos del soporte impreso al que más se adapte a nuestras necesidades. En los viejos tiempos eso significaba pasarse horas leyendo delante de un grabador o subrayando con lápices de colores, ahora solamente necesitamos convertir el archivo en formato gráfico que obtuvimos en el escaneo a formato texto y luego convertirlo a mp3 o crear presentaciones con *Impress*.

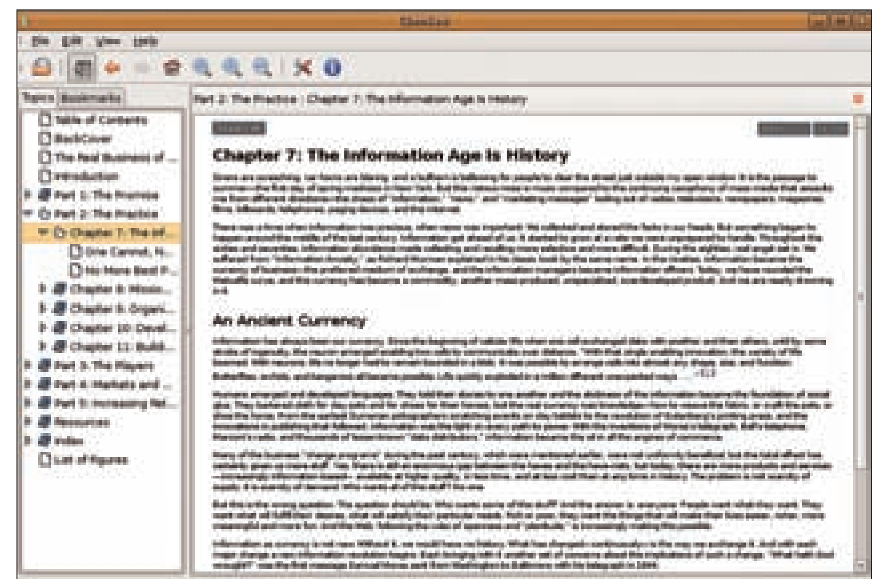


Figura 2. ChmSee

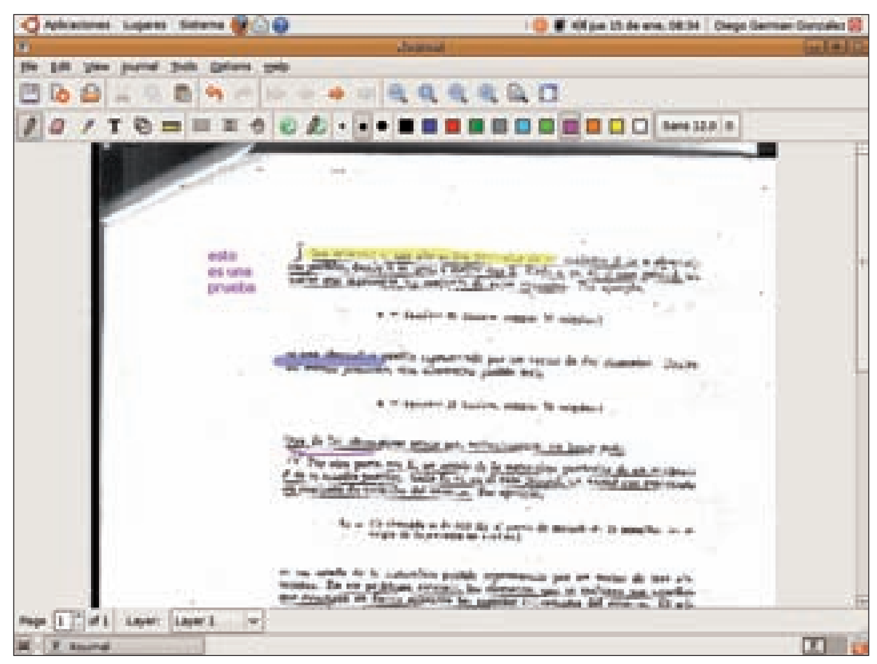


Figura 3. Xournal

Hasta hace un tiempo estaba convencido de que debía optar entre usar Linux o poder tener un buen software de OCR, investigando para este artículo descubrí que manipulando la imagen se obtenían buenos resultados, pero todavía estamos lejos de la calidad y facilidad de uso de alternativas comerciales para Windows como el Abby Finereader.

Una de las herramientas para reconocimiento de texto disponible en los repositorios es GOCR, si bien es posible lanzarlo desde XSANE y hay algunas GUI en etapa de desarrollo sus mejores potencialidades se logran utilizándolo desde la terminal.

Metodología de estudio con el ordenador

A título de ejemplo vamos a proponer una metodología de estudio con la computadora que no es más que la adaptación de las que pueden encontrarse en diferentes lugares para trabajar con material en formato papel, esta metodología consta de las siguientes fases:

- Lectura Previa,
- Lectura Detallada,
- Reorganización del material.

Lectura previa

Esta etapa es la que constituye nuestra primera aproximación al material de estudio, y su objetivo es formarnos una idea general de su contenido. Es recomendable prestar atención a los títulos, subtítulos, epígrafes y estructura del índice. Dependiendo del formato en el que esté el archivo que vamos a leer podremos optar por alguno de estos programas:

- *Evince*: es un visor de documentos en formatos pdf, postscript, djvu, tiff y dvi. Entre sus funcionalidades están la de búsqueda de palabras, la exhibición de índices, la muestra de páginas en miniatura y la funcionalidad de copiar/pegar para editar los párrafos seleccionados en un procesador de textos.
- *Chmsee*: mucho material de estudio sobre informática puede encontrarse en la web en formato chm, un estándar creado por Microsoft para los archivos de ayuda de Windows. Chmsee nos permite ir directamente a una página del índice que nos muestra a la izquierda o ir página por página como si estuviéramos leyendo un libro o una revista, también podemos ampliar o reducir el formato del texto.
- *OpenOffice.org Writer*: cuesta acostumbrarse, pero cuando lo hacemos, leer con esta herramienta es muy fácil. Podemos seleccionar el tipo y color de la letra y el fondo de pantalla y ponerlo a pantalla completa para leer con mayor comodidad.

Lectura detallada

En este apartado vamos a utilizar un programa llamado *Xournal* que permite trabajar con archivos pdf como si utilizáramos lápices de colores sobre un libro tradicional.

Para trabajar disponemos de un lápiz, un borrador, un resaltador de textos, un recuadrador (para ser más exactos un trazador de polígonos) y una herramienta de textos que nos permite hacer nuestras anotaciones al margen. Podemos optar por tres tamaños de trazo y 11 colores diferentes.

Una posibilidad interesante es la de repetir la página actual para poder trabajar sobre la copia o la de adicionar una hoja en blanco para poder hacer nuestras anotaciones con más comodidad. Podemos hacerlo de la siguiente manera:

- Vamos a la pestaña *File* y seleccionamos *Annotate pdf*.
- Vamos a la pestaña *Xournal* y elegimos alguna de estas opciones:
  - *New page before*: coloca una copia de la página actual antes de ésta.
  - *New page after*: coloca una copia de la página actual después de ésta.
  - *New page at end*: coloca una copia de la página actual al final del texto.
  - *Paper style*: agrega una hoja en blanco con alguno de estos estilos (*Plain*: completamente en blanco; *Lined*: con renglones y margen; *Rules*: con renglones pero sin margen; *Graph*: coloca una hoja cuadriculada).

Trabajando con las herramientas de *Xournal* podemos lograr las siguientes ventajas:

- Una más fácil comprensión de la estructura y organización del texto.
- Lograr fijar la atención y el interés en lo que estamos haciendo por más tiempo.
- Al vernos obligados a diferenciar lo principal de lo secundario ejercitamos nuestro sentido crítico.
- Necesitamos menos tiempo para repasar lo estudiado.
- Nos resulta más fácil la elaboración de esquemas y resúmenes.

Ejemplo de utilización del programa

Con la herramienta Pen o la herramienta Ruler subrayamos con rojo la idea principal de cada párrafo y con verde las secundarias. Con la herramienta Highlighter resaltamos en amarillo las palabras claves del texto. Con la herramienta Text escribimos en el margen preguntas cuyas respuestas sean las ideas que estamos marcando.

Reorganización del material

La última etapa en la fijación del conocimiento es el reordenamiento de lo leído, entendiendo por ésto la realización de esquemas en los que de manera fácil podamos ver cómo se relacionan los diferentes conceptos entre sí. En la realización de los mismos debemos expresar las ideas subrayadas en frases breves y utilizando en lo posible las palabras claves resaltadas.

Con *Xournal* podemos realizar de manera muy fácil los esquemas:

- Vamos a la pestaña *Journal* y seleccionamos *Paper size*, elegimos como tamaño A4 Landscape.
- Seleccionamos *Paper style* y elegimos *Plain*.
- Si lo deseamos podemos cambiar el color de fondo con *Paper color*.

En ésto nos resultará de particular utilidad la herramienta *Select rectangle* que nos permite mover porciones de la página a otros lugares de la misma.

Trabajando con archivos multimedia

Lentamente las nuevas tecnologías van transformando la oferta de materiales educativos, por un lado los teléfonos celulares incorporan funciones de grabación de audio y vídeo y por el otro el aumento de las velocidades de Internet facilita el intercambio de archivos multimedia. En este apartado comentaremos algunos programas de utilidad para el trabajo con este tipo de archivos.

Conversión de formatos

Supongamos que solamente nos interesa el audio de una conferencia grabada en vídeo o queremos poner a disposición de nuestros alumnos un documental sobre Sudáfrica. Un software imprescindible para esto es Winff.

El método de trabajo es sencillo:

- Con la pestaña *Añadir* seleccionamos los archivos a convertir.
- Con la pestaña *Convertir A* seleccionamos el tipo de dispositivo con el que vamos a utilizar el archivo convertido y en la pestaña de al lado el formato de destino.
- Con la pestaña *Convertir* damos inicio al proceso.

Animaciones interactivas

En la web existen numerosas aplicaciones interactivas desarrolladas en formato swf, que se pueden ver sin problemas con nuestro navegador, pero si deseamos descargarlas a nuestro disco rígido podemos visualizarlas con gnash.

Audios educativos

Cuando grabamos una clase o una conferencia no siempre podemos asegurarnos de que la calidad del sonido obtenido sea la mejor. Por suerte trabajando con herramientas como AUDACITY podemos mejorarlo sustancialmente. Una de las cosas que podemos hacer con este programa es eliminar los bajos que saturan la grabación:

- Abrimos el archivo con la pestaña *Archivo abrir*.
- Vamos a *Efectos complementos 1 a 7 high pass filter* y ponemos cutoff frequency en 150 khz.

También podemos eliminar el ruido, lo que se hace de la siguiente forma:

- Seleccionamos un sector del archivo que contenga únicamente ruido (lo hacemos con un clic en el botón izquierdo y arrastrando el puntero del ratón)
- Vamos a *Efectos reducción de ruido* y hacemos clic en *Obtener perfil de ruido*.

Otro filtro de Audacity que puede resultarnos de utilidad es el que permite amplificar el volumen de una parte del audio:

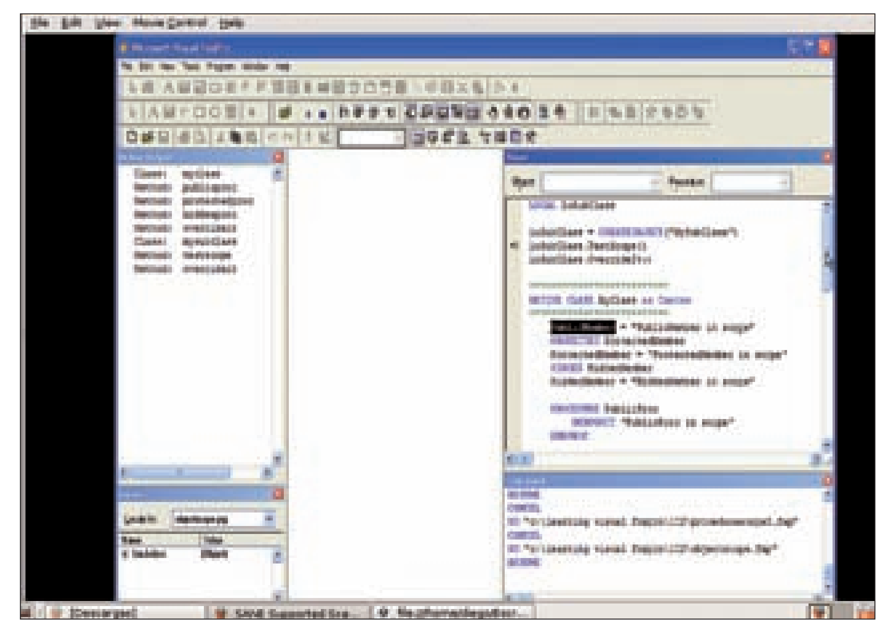


Figura 4. Gnash



En la red

- *http://www.sane-project.org/*
- *http://www.xsane.org/*
- *http://unpaper.berlios.de/*
- *http://jocr.sourceforge.net/*
- *http://xournal.sourceforge.net*
- *http://code.google.com/p/winff/*
- *http://audacity.sourceforge.net/*

- Seleccionamos con el ratón la porción del audio que nos interesa y hacemos clic en el efecto *Amplificar*.

Videos educativos

A partir de la extensión de la televisión por cable con sus canales de documentales, la aparición de portales de streaming y la baja del precio de las cámaras digitales, tenemos a nuestra disposición una gran cantidad de vídeos educativos que constituyen una importante base de conocimientos. A nuestro criterio una de las mejores aplicaciones a la hora de trabajar con ellos es *Kino*.

Kino trabaja nativamente con el formato DV, si nuestros archivos están en otro formato el programa se encarga de convertirlos antes de empezar a trabajar.

Kino nos permite visualizar el vídeo en forma de fotogramas lo que facilita la eliminación de tramos, exportar determinados fotogramas para editarlos con un programa gráfico (por ejemplo poner una flecha que señale determinado aspecto de la imagen, modificar la banda de sonido, poner subtítulos y unir varios vídeos). El resultado final puede exportarse a los formatos de vídeo más conocidos

Palabras finales

En 1986 cuando comencé a interesarme por la informática se escribían artículos explicando cómo la tecnología de las computadoras iba a revolucionar la educación. Veintidós años después la revolución se limita a la inclusión de materias en las que se enseña a usar los programas de Microsoft y a alumnos que copian y pegan sus tareas escolares de páginas de Internet.

Dentro de la comunidad Linux existen múltiples proyectos destinados a desarrollar herramientas de aprendizaje, aún cuando es mucho lo que queda por hacer en materia de facilidad de uso y prestaciones lo que ya tenemos es más que suficiente como para empezar a experimentar formas creativas de enseñanza y aprendizaje que aprovechen al ordenador en todas sus potencialidades.

www.buyitpress.com

¡Suscríbete a tus revistas favoritas
y pide los números atrasados!

¡Regalos para nuevos suscriptores!

Ahora te puedes suscribir a tus revistas preferidas en tan sólo un momento y de manera segura.

Te garantizamos:

- precios preferibles,
- pago en línea,
- rapidez en atender tu pedido.

¡Suscripción segura a todas las revistas de Software-Wydawnictwo!

Pedido de suscripción






Por favor, rellena este cupón y mándalo por fax: 0048 22 244 24 59 o por correo: Software-Wydawnictwo Sp. z o. o., Bokserska 1, 02-682 Varsovia, Polonia; e-mail: *suscripcion@software.com.pl*

Nombre(s) .... Apellido(s) ....

Dirección ....

C.P. .... Población ....

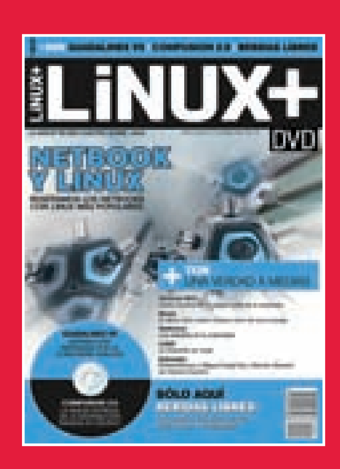
Teléfono .... Fax ....

Suscripción a partir del Nº ....

e-mail (para poder recibir la factura) ....

☐ Renovación automática de la suscripción







Título	número de ejemplares al año	número de	a partir del número	Precio
Linux+DVD (1 DVD) Mensual con dos DVDs dedicado a Linux	12			69 €

En total

Realizo el pago con:

☐ tarjeta de crédito (EuroCard/MasterCard/Visa/American Express) nº ____ ____ ____ ____ CVC Code ___

Válida hasta ____

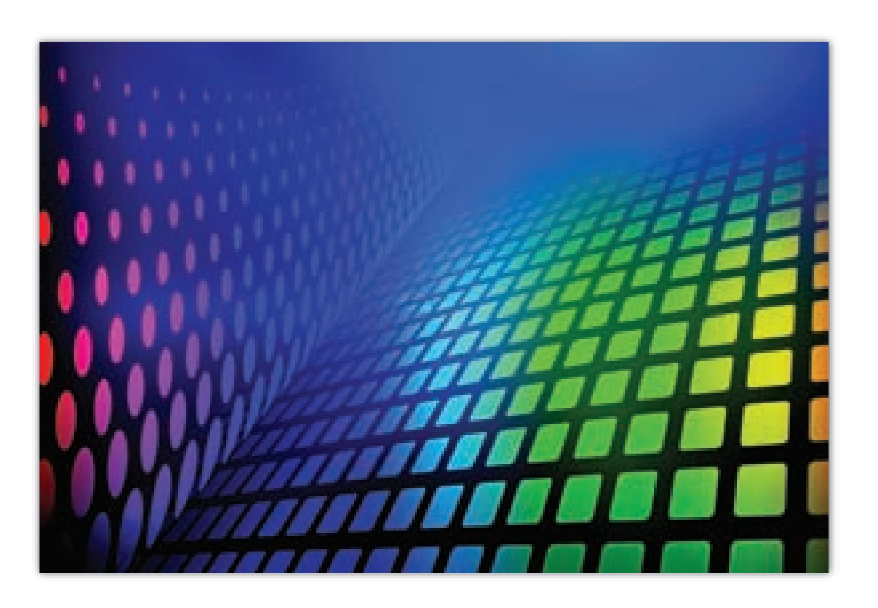
☐ transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO
Número de la cuenta bancaria: 0049-1555-11-221-0160876
IBAN: ES33 0049 1555 1122 1016 0876
código SWIFT del banco (BIC): BSCHESMM

Fecha y firma obligatorias:

Jotahacker:
Programando un gestor de paquetes universal

Juan Carlos de la Vega Salamanca

Uno de los proyectos más ambiciosos que se me podrían haber ocurrido, si me lo hubiese preguntado alguien, hubiese sido el de un instalador universal para GNU/Linux.



linux@software.com.pl

Lo hubiera elegido precisamente porque tanto para las personas que usan habitualmente este sistema operativo como, con más razón, para los recién llegados, la instalación de programas en GNU parece, a día de hoy, una de las cuestiones pendientes más importantes para estandarizar el sistema en los escritorios de todo el mundo.

Introducción

Dentro de los gestores/instaladores de paquetes *synaptic* es para algunos el rey y *emerge* es Dios para otros. Yum, urpmi, pacman, hay muchos, manejables desde terminal o a través de una interfaz gráfica que, aunque permita menos control, ayude a los menos *geeks*. Por fin también se puede compilar directamente desde el código fuente del programa empaquetado para optimizar la instalación. Todo dependerá, en definitiva y salvo en caso de la compilación, de la distribución del sistema que estemos utilizando. Por formas de instalar un programa en GNU/Linux no será, hay casi para todos los gustos. Para todos menos para uno, el de aquellos que preferirían que en GNU/Linux se pudiese instalar el mismo programa de la misma forma en cualquiera de sus distribuciones. El problema para muchos, sobre todo gente que no se ha destetado en este sistema y entre los que me incluyo, que provenimos de otros sistemas como Windows o Mac OS, como es mi caso, es que el único sistema *universal* que existe, el de la compilación, plantea el problema de la resolución de las dependencias, es decir, qué paquetes auxiliares y en qué orden instalar para que el programa elegido se ejecute sin ningún problema.

Pues resulta que existe un proyecto español que tiene mucho que ver con todo lo dicho hasta ahora que parece destinado a convertirse en el instalador estándar usado por todos, la inmensa mayoría en todo caso, en no demasiado tiempo y que puede significar un gran avance hacia la unificación de todo este mundillo *disperso* del GNU que puede contribuir a auparlo hasta el primer puesto de los sistemas operativos de escritorio y empresariales, comerciales o no, más usados del mundo. Se llama *Nhopkg*.

Jotahacker, creador de Nhopkg

Ahora un poco de historia, primero sobre el *hacker* a cargo de la tarea: Jotahacker.

Jotahacker es el *nick* de un chico nacido el 24 de Septiembre de 1991, normal y corriente (o no tanto), de Guadalajara (Es-



Figura 1. Logo del proyecto nhopkg

paña) que se llama Jaime Gil de Sagredo Luna, según sus propias palabras *seguidor y defensor del Software Libre*, aficionado a la informática y a casi todo lo que esté relacionado con las nuevas tecnologías. Estudia en uno de los institutos públicos de esta localidad 2º de Bachillerato por la rama de Ciencias de la Naturaleza y de la Salud y, como premio a su esfuerzo y valía, obtiene unas respetables buenas notas y tiene previsto estudiar Ingeniería Informática Superior.

Quizá estemos hablando, sin saberlo con certeza a día de hoy, de una de esas personas llamadas a cambiar radicalmente el panorama tecnológico, y eso significaría la vida, de todos nosotros a no mucho tardar. Tiempo al tiempo, que se dice en estos casos.

Es decir, estamos hablando de que es un brillante joven de 17 añazos el que ha emprendido un proyecto como éste con la modestia, sinceridad, claridad de ideas y, por qué no, locura que sólo manifiestan aquellos llamados a llevar a cabo grandes empresas, demasiado gravosas y complicadas para el común de los mortales. Suena como es.

Jotahacker es autodidacta y sus conocimientos informáticos sobre ordenadores, sistemas operativos y programación los aprendió, los aprende cada día, delante de un ordenador en su casa, en su habitación.

El *gusanillo* de la programación le picó hace como tres o cuatro años, antes de que supiera siquiera que existía un sistema llamado GNU. Empezó programando en VB.NET y Visual Basic. Hacía sus pinitos en estos dos lenguajes de programación *porque era lo que tenía más a mano*. Más tarde, ya con GNU/Linux, fue aprendiendo algo de PHP, SQL, y varios lenguajes más aunque de forma no muy estricta. El último lenguaje que aprendió fue Bash, el shell scripting, y es con éste con el que más ha trabajado porque, aunque sencillo y quizá mucho menos avanzado que otros, siempre ha servido bien a sus necesidades. Nhopkg, por ejemplo y para no ir más lejos, está programado en Bash.

Empezó a usar sistemas operativos GNU/Linux en sus ordenadores hace unos dos años más o menos. La razón fue que durante el verano se quedó sin su ordenador y no podía esperar a conseguir otro, por lo que se decidió a devolver a la vida su viejo PC (Intel Celeron 300 Mhz y 64 Mb de RAM) intentando instalar en esta máquina una distribución de GNU/Linux. Conocía algo sobre que existían distribuciones ligeras de este sistema para PCs con pocos recursos, de modo que se hizo con varias y fue probando. Al final todo funcionó a la perfección, incluida la tarjeta WIFI, que fue lo que más le costó. A partir de entonces, cuando por fin consiguió su ordenador nuevo, siempre ha usado GNU/Linux en todos sus ordenadores. Empezó con Ubuntu y actualmente usa Fedora.

El primer proyecto que abordó y del que derivan todos los demás fue la creación de una distribución de GNU/Linux. Una cosa tenía clara por aquel entonces, no estaría basada en ninguna de las existentes (Debian, Ubuntu, Fedora...) sino que partiría de cero para crear algo *nuevo y diferente*. Con esa premisa inicial se decantó por crear una distribución LFS (Linux From Scratch), ayudado por la gran cantidad de información que facilitan. Tanto ímpetu no era recompensado por aquel entonces con resultados visibles y los problemas y dificultades se multiplicaban mientras *Jotahacker* buscaba una versión definitiva para lanzar su distro. Según sus propias palabras, *al ser un proyecto tan poco pensado y meditado (al contrario de como debería haber sido) siempre había errores muy básicos que le impedían llegar a liberar la versión definitiva*. Uno de los problemas con los que se encontraba era la falta de un gestor de paquetes que facilitase la instalación y desinstalación de software para reemplazar fácilmente aquellos paquetes que resultaran problemáticos o poder probar diferentes versiones de un mismo paquete. Así surgió *Nhopkg*.

Nhopkg

Nhopkg nació como un pequeño script en Bash destinado a instalar y desinstalar paquetes binarios para su proyecto de distribución LFS con el fin de facilitar en la medida de lo posible la gestión de software a los usuarios finales, empezando por él mismo. Su primera versión, la 0.1, fue liberada bajo la licencia GNU GPL v2 a través de *Sourceforge.net* que aprobaron el proyecto y contaba con apenas 122 líneas de código (la 0.4 tiene ya 737 líneas de código) enteramente escrito por Jotahacker. De momento no ha utilizado en su programa líneas de código escritas por otros, como pudiera haber hecho aprovechando la licencia GPL y la libertad que ésta concede en este aspecto, con el fin de simplificar y acelerar la creación de software de calidad y salvando, rescatando en otros casos, el código de los taimados secretismos y de las apropiaciones perpetradas por las multinacionales del software privativo.

Nhopkg toma su nombre de dos fuentes. La primera es el proyecto en el que iba a ser incluido, que se iba a llamar *nhoax*, la segunda es el característico *pkg* que hace alusión a los paquetes de software. De la primera se toma el *Nho*, de la segunda el *pkg* y ¡listo!, un nombre con todo el sabor de Unix, sí señor.

Las necesidades fueron surgiendo, había que añadir a Nhopkg la funcionalidad de crear paquetes binarios a partir del código fuente de los programas. Jotahacker empezó a trabajar en ello desde la versión 0.2 sin que ésta aún incluyese la mejora. En la versión 0.3 decidió que los binarios se crearían a partir de paquetes *fuente* específicos para Nhopkg. Esta versión contó con ciertas mejoras y corrección de errores pero no sería hasta la versión 0.4, la actual versión estable, cuando se añadió por fin esta posibilidad. También se añadió un asistente para la creación, a partir de un paquete de *fuentes* convencional (`.tar.bz2` o `.tar.gz`) un paquete de fuentes para Nhopkg (`-src.nho`). Sin lugar a dudas

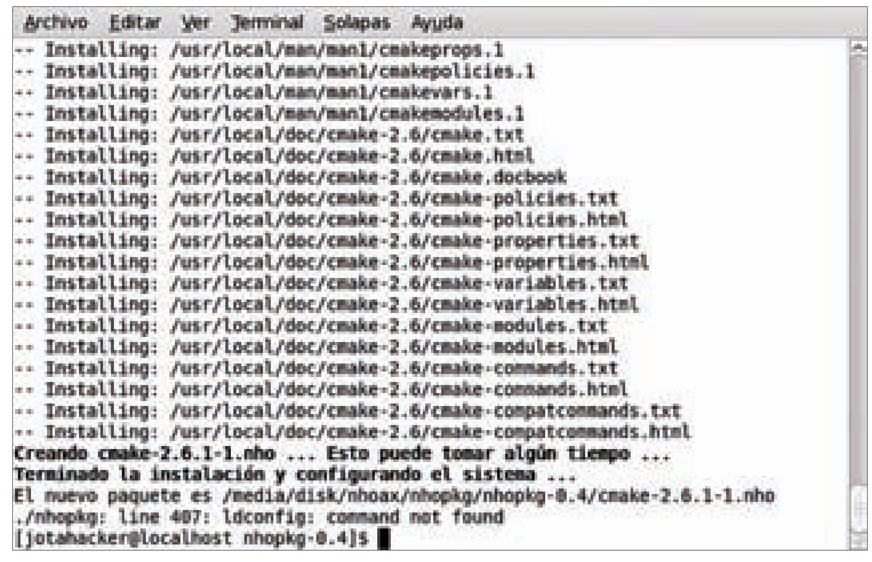


Figura 2. Gestor de paquetes, trabajando sobre Fedora (1)

fue esta versión, la 0.4, la que mayor número de nuevas funcionalidades y mejoras ha sufrido a la espera, eso sí, de la 0.5 prevista para Febrero de 2009 y fue a partir de este momento, el lanzamiento de la versión 0.4, cuando el proyecto alcanzó la madurez suficiente para que Jotahacker se diese cuenta que podía aspirar a ciertas metas que en un comienzo ni se le hubiese ocurrido plantearse, como es transformar su pequeño y modesto script en bash en un gestor de paquetes universal para ser usado fácilmente por todo el mundo y en cualquier distribución, no sólo por él y en la suya propia.

Con la versión 0.4 de Nhopkg su primer proyecto, la creación de una distribución LFS, despertó de nuevo al poder construir de forma completa un sistema LFS básico sin ningún problema, no siendo el único que lo ha hecho hasta la fecha, pues a pesar de la corta vida del programa ya hay un proyecto libre, también basado en LFS, al que Nhopkg se adapta muy bien, que lo utiliza como gestor de paquetes por defecto. Se trata de la distribución venezolana *Vegnux*.

La versión 0.4 también ha significado un enorme salto mediático para Jotahacker y su Nhopkg, pues en webs y blogs se han multiplicado los artículos hablando de él y de su notorio gestor de paquetes (así nos conocimos). Propuestas para participar, e incluso unirse, en otros proyectos o la aparición de voluntarios para traducir el programa a otros idiomas han sido sólo algunos de los *efectos colaterales* de este miniestallido mediático en Internet.

Actualmente Jotahacker está trabajando en la versión 0.5 e incluso por primera vez ha anunciado una fecha de lanzamiento, el 1 de Febrero de 2009 (que significa que a la hora de leer este artículo ya podéis disfrutar de él). Muy al estilo Debian este lugar en el calendario es tan sólo una referencia y se puede posponer el lanzamiento de la última versión del gestor de paquetes, puesto que prima la corrección de errores y el dejar el programa lo más hilvanado posible frente a la exactitud en el plazo de entrega. Lo importante es que contenga el mínimo número de errores posible.

En esta versión se saldará una cuenta pendiente que aún no se había saldado en anteriores versiones, la gestión de las dependencias, algo que facilita muy mucho a cualquier usuario la transición al sistema GNU desde otros sistemas operativos donde no existe el concepto de *dependencia*. La 0.5 cuenta con un gestor de dependencias que nada tiene que envidiar a otros más conocidos y comúnmente utilizados en las principales distribuciones de GNU/Linux sobre el planeta que indagará en nuestro sistema buscando si cumple todos los requisitos para la perfecta instalación y ejecución de cualquier programa que deseemos instalar y, caso de no encontrarlo todo en orden, nos ayudará a solventar sencilla y muy rápidamente la situación. No olvidemos que al ser universal Nhopkg hará todo este trabajo por nosotros sobre cualquier distribución.

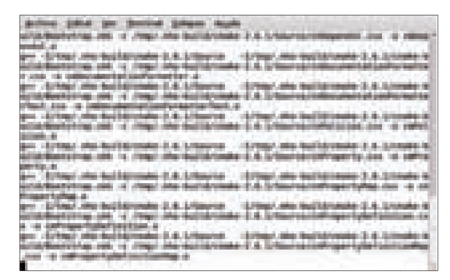


Figura 3. Gestor de paquetes trabajando sobre Fedora (2)

Otra mejora es la significativa reducción del tiempo que Nhopkg tarda en crear los paquetes binarios. A partir de esta versión no habrá que esperar demasiado incluso con los paquetes más pesados. También tendremos, a partir de ahora, información sobre el paquete que estemos instalando, como puede ser descripción sobre el paquete, licencia usada, arquitectura del procesador, fecha de instalación, etc. Habrá un cambio de licencia, de la versión 2 de la GPL con que están licenciadas las versiones anteriores a la versión 3 de dicha licencia. Habrá muchas pequeñas mejoras que apenas se notarán *desde fuera* pero que aumentarán el rendimiento y la eficacia del programa. Por ejemplo, aunque todavía no está decidido si entrará en esta última versión (0.5), Jotahacker está trabajando en una opción (en realidad serán varias) para que desde el mismo Nhopkg se puedan transformar paquetes `.deb`, `.rpm` y `.tgz` (usados por otros gestores de paquetes) a los paquetes `.nho` sin necesidad de tener instalado ni dpkg, ni rpm, ni, en definitiva, sus correspondientes gestores de paquetes. Esto ya lo hace *alien* pero para ello necesita que estén instalados *rpm* y *dpkg*, detalle que nos revela la importancia y novedad de esta *pequeña mejor*a.

Los proyectos de futuro de este excepcional estudiante de Bachillerato están liderados, y supeditados, al desarrollo y evolución de Nhopkg. De él depende el futuro lanzamiento de su distribución, que será 100% libre, y el desarrollo de un *gran proyecto* que se apoya en los dos anteriores y del cual de momento no desvelaremos nada, por lo imprevisible de su desarrollo, salvo que estará ligado a la divulgación del Software Libre.

Conclusión

Poco queda ya por contar sobre uno de los proyectos más interesantes y cercanos que un joven y brillante hacker nos brinda con esa fuerza que dan el genio y la juventud a partes iguales. Cualquier persona que esté interesada en colaborar con Jotahacker o simplemente obtener más información o que esta sea de primera mano, revelada por el propio autor del proyecto, que no dude en pasar por la web del proyecto (*http://Jotahacker.es/*) y ponerse en contacto con él.

Jotahacker es consciente de que por delante hay mucho más trabajo que lo ya hecho, muchas dificultades, más de las imaginables, tanto técnicas como humanas que sembrarán el desarrollo de algo tan sonoro y con tanta proyección como este gestor de paquetes. Preguntas como si reaccionarán bien los usuarios a su propuesta o si intereses ajenos al desarrollo de software intentarán impedir la implantación de Nhopkg en las grandes distribuciones sólo pueden contestarse día a día, con el paso del tiempo. Qué duda cabe que la suerte, esa manera tan humana de intentar poner de nuestro lado el complejo sistema de impredecibles acontecimientos futuros, es quizá el factor más determinante de todos a la hora de sacar adelante un proyecto tan brillante y ambicioso. Si es suerte lo que necesita suerte le deseo, porque no me cabe duda que con todo lo demás sabrá lidiar.

Para terminar, mi más sincero agradecimiento a la persona, Jaime Gil de Sagredo Luna, que ha sido desde el primer momento en que los bits comenzaron a fluir entre él y yo una persona amable y cercana que no ha dudado en ofrecerme toda la información que le haya podido solicitar para la redacción de este artículo, facilitándome con ello la tarea enormemente.

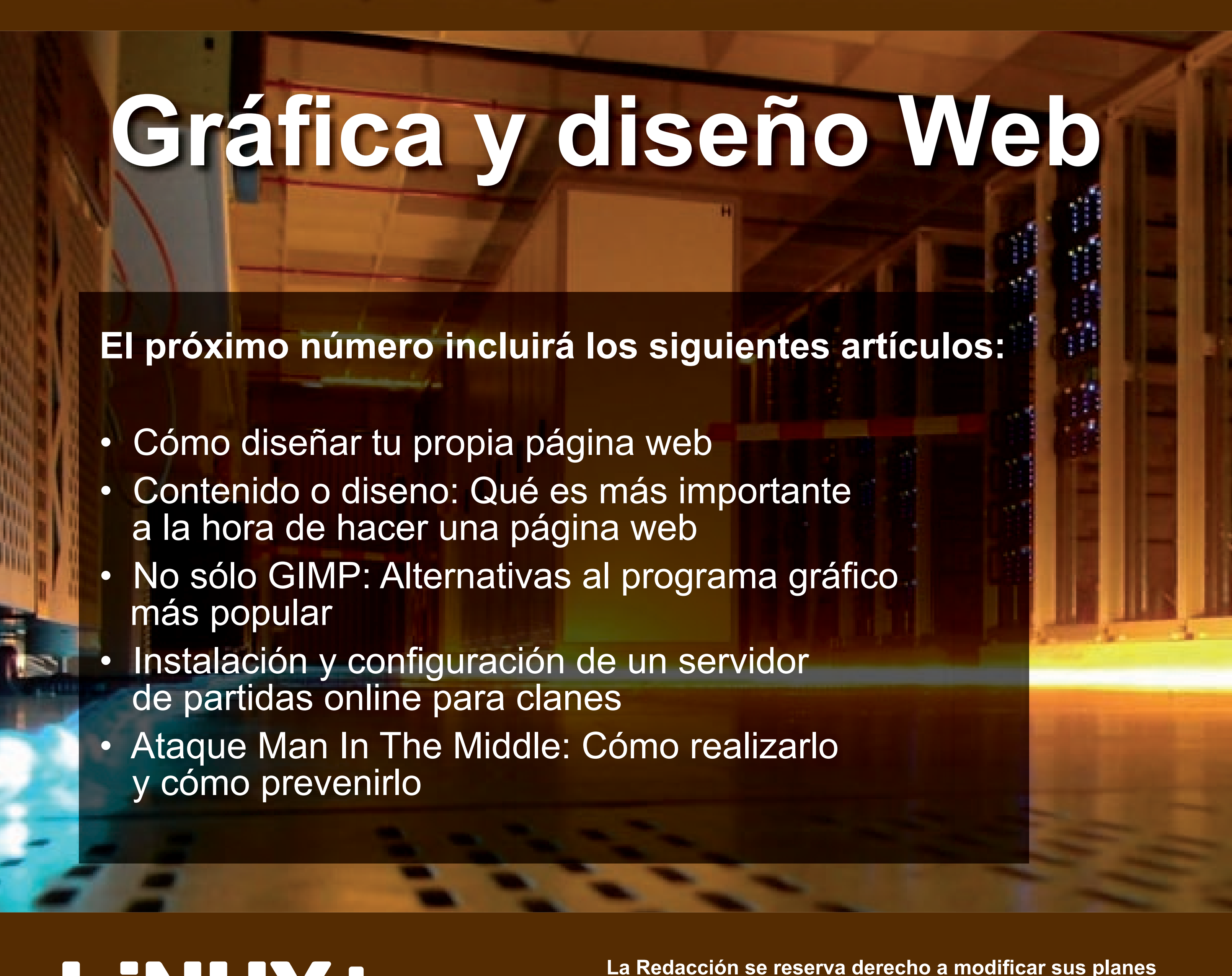
A veces, por fortuna, son excepcionales el genio y la persona; es entonces cuando el mundo comienza a brillar.



Sobre el autor

Me llamo Juan Carlos de la Vega Salamanca, nací en León (1972). De formación científica y vocación equivocada soy ante todo poeta y fotógrafo y he desarrollado toda mi actividad en la ciudad de León (España). Desde hace no mucho que me incorporé a la Web 2.0 también divulgador de la conveniencia del uso del Software Libre en todos los ámbitos de la cotidianeidad informática a través del blog PILLATEUNLINUX (*http://pillateunlinux.wordpress.com*) del que soy coadministrador y redactor. También me dedico a ayudar a los más novatos en el sistema GNU con pequeños tutoriales que publico en el blog PUTOLINUX (*http://putolinux.wordpress.com*), concebido como unos apuntes personales que pongo a disposición del que lo necesite, y en el canal de IRC-Hispano #pillateunlinux.

El tema principal del siguiente número de Linux+ será:



Gráfica y diseño Web

El próximo número incluirá los siguientes artículos:

- Cómo diseñar tu propia página web
- Contenido o diseno: Qué es más importante a la hora de hacer una página web
- No sólo GIMP: Alternativas al programa gráfico más popular
- Instalación y configuración de un servidor de partidas online para clanes
- Ataque Man In The Middle: Cómo realizarlo y cómo prevenirlo

LiNUX+

La Redacción se reserva derecho a modificar sus planes

Mensual Linux+ está publicado por Software-Wydawnictwo Sp. z o. o.

Producción:
Marta Kurpiewska, *marta@software.com.pl*
Product Manager
Paulina Pyrowicz, *paulina.pyrowicz@software.com.pl*
Redactora jefe:
Aleksandra Tęsiorowska, *a.tesiorowska@lpmagazine.org*
Colaboradores:
Francisco Javier Carazo Gil, José Carlos Cortizo Pérez, Osvaldo Rodolfo Salazar Sánchez, Sebastián Pignataro

Correctores:
Pablo Cardozo, Roberto Delgado Sánchez

Preparación de DVDs:
Ireneusz Pogroszewski, Andrzej Kuca

DTP:
Marcin Ziółkowski
Graphics & Design Studio, *www.gdstudio.pl*

Diseño portada:
Agnieszka Marchocka
Gráfico:
Łukasz Pabian – "insane"
Publicidad:
adv@software.com.pl
Suscripción:
suscripcion@software.com.pl

Distribución:
Coedis, S. L.
Avd. Barcelona, 225
08750 Molins de Rei (Barcelona), España

Dirección:
Software–Wydawnictwo Sp. z o.o.,
ul. Bokserska 1,
02-682 Varsovia,
Polonia

La Redacción se ha esforzado para que el material publicado en la revista y en los DVDs que la acompañan funcionen correctamente. Sin embargo, no se responsabiliza de los posibles problemas que puedan surgir.

Todas las marcas comerciales mencionadas en la revista son propiedad de las empresas correspondientes y han sido usadas únicamente con fines informativos.

La Redacción usa el sistema de composición automática AUPUS

Los DVDs incluidos en la revista han sido comprobados con el programa AntiVirenKit, producto de la empresa G Data Software Sp. z o.o.

¡Advertencia!
Queda prohibida la reproducción total o parcial de esta ublicación periódica, por cualquier medio o procedimiento, sin para ello contar con la autorización previa, expresa y por escrito del editor.

Linux ® es una marca comercial registrada de Linus Torvalds.

Encuesta

Datos Personales

Género ....

Edad ....

Formación ....

Domicilio:

- ☐ pueblo <5.000
- ☐ ciudad 5.000 - 100.000
- ☐ ciudad 100.000 - 500.000
- ☐ ciudad >500.000

Sueldo anual:

- ☐ <10.000
- ☐ 10.000 - 20.000
- ☐ 21.000 - 50.000
- ☐ >51.000

Preguntas

¿Con qué frecuencia utilizas ordenador? ....

¿Con qué frecuencia utilizas internet? ....

¿Qué sistema operativo usas? ....

- ☐ MS Windows
- ☐ MAC OSX
- ☐ GNU / Linux
- ☐ otro ....

¿Qué sistema usas en trabajo? ....

¿Qué sistema usas en casa? ....

¿Desde cuándo usas GNU / Linux? ....

¿Para qué usas GNU / Linux? ....

¿Para qué usas otros sistemas operativos? ....

¿En qué equipo usas GNU / Linux? ....

- ☐ Portátil
  Parámetros ....
  ....
- ☐ Ordenador fijo
  Parámetros ....
  ....

Cinco palabras con las que describirías GNU / Linux:

....

....

....

....

....

Encuesta

¿Qué revista(s) lees?

Revista	Nunca	A veces	A menudo	Siempre
@rroba	☐	☐	☐	☐
Computer World	☐	☐	☐	☐
iWorld	☐	☐	☐	☐
Linux Magazine	☐	☐	☐	☐
Linux+ DVD	☐	☐	☐	☐
MacWorld	☐	☐	☐	☐
PC Actual	☐	☐	☐	☐
qwerty	☐	☐	☐	☐
Sólo programadores	☐	☐	☐	☐
Otra	☐	☐	☐	☐

Esta sección la leo:	Nunca	A veces	A menudo	Siempre
Linux en la empresa	☐	☐	☐	☐
Noticias	☐	☐	☐	☐
Seguridad	☐	☐	☐	☐
Software	☐	☐	☐	☐
Multimedia	☐	☐	☐	☐
Práctica	☐	☐	☐	☐
Hardware	☐	☐	☐	☐
Proyectos Linux	☐	☐	☐	☐
Linux en la educación	☐	☐	☐	☐
Internet	☐	☐	☐	☐
Conocimiento libre	☐	☐	☐	☐
Trucos y truquillos	☐	☐	☐	☐
Socorro	☐	☐	☐	☐
Juegos	☐	☐	☐	☐
Data Center	☐	☐	☐	☐
Virtualización	☐	☐	☐	☐
Para programadores	☐	☐	☐	☐
Entrevista	☐	☐	☐	☐
Gráfica	☐	☐	☐	☐
Test de consumidores	☐	☐	☐	☐
Hosting	☐	☐	☐	☐
Para principiantes	☐	☐	☐	☐
Descripción de DVD	☐	☐	☐	☐

Comentarios

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

¿Por qué tomar parte en la encuesta?

La encuesta nos aportará información básica sobre vosotros, gracias a lo cual podremos ajustar el perfil de la revista a vuestros intereses y expectativas.

Gracias a ella podrás ganar:

- 2 Bluetooth USB Dongle
- 1 ratón óptico
- 1 PC cámara
- 3 cubiertas protectoras Gelaskins para portátil
- 3 cubiertas protectoras Gelaskins para iPod
- 4 kits de limpieza de pantalla

Si quieres tomar parte en la rifa y obtener regalo, necesitaremos tu email y dirección:

email ....

Nombre y Apellidos ....

Calle ....

Población ....

CP ....

País ....





Sigue la evolución de los proyectos desde:

www.concursosoftwarelibre.org/planet



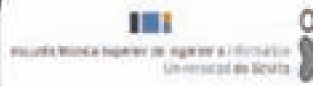
Organiza



Colaborador Principal

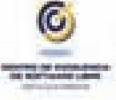


Universidades Colaboradoras





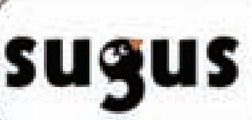






Organismos Colaboradores







Medios oficiales







Think smart

ESET

Smart

Security

Un nuevo concepto en protección inteligente para tu PC

Seguramente ya estarás confiando en una suite de seguridad. Hay muchas de ellas, pero sólo ESET ofrece una solución unificada completamente diferente.

Puede pensar.

Gracias a su tecnología Threatsense© tiene la habilidad de anticiparse a peligros potenciales, sin ralentizar tu sistema operativo y protegiendo proactivamente tu ordenador.

Es inteligente.

Sé también proactivo y prueba tu versión de evaluación gratuita de 30 días en www.eset.es

COMPONENTES INTEGRADOS
ESET NOD32 Antivirus
ESET NOD32 Antispyware
ESET Personal Firewall
ESET Antispam







c/Martínez Valls 56, bajos - 46870 Ontinyent (Valencia)
ventas@protegerse.com - Teléfono 902.33.48.33
http://www.eset.es